Installing and Using HP BASIC in the MS-DOS Environment

HEWLETT PACKARD

HP Part No. 82324-90000
Printed in USA

Notice

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

Hewlett-Packard 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. Hewlett-Packard 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.

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

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

Printing History

First Edition — June 1989
Second Edition — November 1990

Contents

3. Getting Started with HP BASIC

C. Selected Graphics CSUBs

D. Language Extension Binaries

E. Error Messages

F. BASIC.EXE Options and the Boot Search Order

Index

1

Introduction

Hewlett-Packard offers two measurement coprocessor products for the PC (personal computer). The HP 82300C Measurement Coprocessor turns your HP Vectra PC or AT-compatible PC into a *multiprocessing computer aided test system*. The HP 82324A High-Performance Measurement Coprocessor also does this, but with more than double the speed for most operations. Both coprocessors use the same software and provide identical HP BASIC functionality, except as noted in this manual.

In this manual we will refer to the "measurement coprocessor" whenever both coprocessors function the same. Where there are differences in functionality, the specific coprocessor will be identified.

Software Compatibility

The measurement coprocessor software is compatible with both the HP 82324A High-Performance Measurement Coprocessor and the HP 82300C Measurement Coprocessor. The software is also compatible with previous versions of the HP 82300 product known as the "HP BASIC Language Processor":

- The HP 82300C Measurement Coprocessor was formerly known as the HP 82300C BASIC Language Processor. The HP 82300C product has been renamed, but this name change involves no changes to functionality. The HP 82300C product, by either name, includes "PC 300" boot ROMs and 1 MB of on-board RAM, and is fully compatible with the current measurement coprocessor software.

- The earlier versions of the product were different in two respects. The HP 82300A/B BASIC Language Processor included "Series 200" boot ROMs and only 512 KB of RAM in the standard configuration. The HP 82300A/B BASIC Language Processor is compatible with the current measurement coprocessor software *provided* at least 1 MB of RAM has been installed.
  If you have an HP 82300A/B BASIC Language Processor with the Series 200 boot ROMs, the procedure for booting the card and starting BASIC is slightly different. (Refer to chapter 3 for further information.)

Software Documentation

The following manuals are included with the measurement coprocessor software:

- *Installing and Using HP BASIC in the MS-DOS Environment.* This manual (which you are now reading) tells you how to install the measurement coprocessor software on your HP Vectra PC or other supported personal computer. It also tells how to start HP BASIC and how to use the HP BASIC programming environment. In addition, it gives techniques for dealing with the MS-DOS environment of your PC.
- *Programming with HP BASIC.* This manual covers selected programming techniques for HP BASIC including general programming techniques, graphics techniques, and interfacing techniques.
- *HP BASIC Language Reference.* This two-volume manual includes a detailed keyword dictionary for the HP BASIC language. It also includes several reference appendixes.
- *HP BASIC Condensed Reference.* This pocket guide, as its name implies, is a condensation of the keyword dictionary.

- *Integrating HP BASIC with MS-DOS Applications*. This manual tells how to integrate your HP BASIC programs with standard MS-DOS applications such as Lotus 1-2-3. This allows you to make use of the measurement coprocessor's *multiprocessing* capability to run an HP BASIC program and an MS-DOS application simultaneously, with direct communication between them.
- *Porting Guide for the Measurement Coprocessor*. This manual tells how to port BASIC programs written for the early-version HP 82300A/B BASIC Language Processor, or for HP 9000 Series 200/300 computers, to run on the HP 82300C Measurement Coprocessor.

There is also a README file on Measurement Coprocessor SW Disk 1, which contains additional information about using your measurement coprocessor. You can read this file with any MS-DOS editor. Or simply insert disk 1 in drive A: and execute:

`TYPE A:\README` (Enter)

from the MS-DOS command line. You can stop the file from scrolling by pressing (Ctrl) (S). (Press any key to resume scrolling.)

Conventions Used in this Manual

Throughout this manual, when you are asked to "execute" an HP BASIC statement, type the statement on the BASIC command line and then press the (Enter) key.

Softkeys (the labels for (F1) through (F8)) are indicated by a dot screen, for example `EDIT`.

Examples of BASIC statements and programs are given in a "computer"-style type. If a statement contains a variable for which you must supply a value, the variable will be in italics, for example:

`PEN` *pen_number*

PC Hardware Requirements

Before you can install the HP BASIC software, you must install your measurement coprocessor in a PC that meets certain hardware requirements. (Actually, you can install up to three measurement coprocessors in your PC, but more about that later.)

You can install your measurement coprocessor in any desk-top or desk-side Vectra PC. (The Portable Vectra CS and Vectra LS models are not supported.) You can also install the measurement coprocessor in an IBM PC-AT computer, an IBM PS-2 Model 30/286 computer, or a 100 percent AT-compatible computer.

Your PC must have at least 640 KB (kilobytes) of internal RAM. (All Vectra PCs have 640 KB or more of internal RAM as standard equipment.) Note that EMS or "expanded" memory does not count as internal RAM.

Your PC must have the MS-DOS operating system installed (version 3.1 or a later version) before you can install the measurement coprocessor software.

Your PC must be equipped with a supported display adapter and monitor. The following display adapters and monitors are supported by the measurement coprocessor:

- Any VGA display adapter and monitor.
- Any EGA display adapter and monitor (at least 256 KB of display RAM is required).
- The HP Multimode display adapter and monochrome monitor.
- Any Hercules-compatible display adapter and monitor.
- The AT&T Monochrome display adapter and monitor.
- The Compaq Portable III plasma display.

Your PC must be equipped with certain internal mass storage devices. However, you have some choices. It is recommended that you install the measurement coprocessor software on an internal hard disk drive. However, you may choose to install the software on a high-density (1.2 MB or 1.44 MB) flexible disk.

For a hard disk installation, your PC must have the following internal mass storage devices:

- One hard disk drive (drive C:).
- One flexible disk drive (drive A:). Either a 5.25-inch (360 KB or 1.2 MB) or a 3.5-inch (1.44 MB) drive may be used.

For a flexible disk installation, your PC must have two internal flexible disk drives. Drive A: is used as the target drive and drive B: is used as the source drive during software installation.

- Drive A: must be a *high-density* flexible disk drive. Either a 5.25-inch (1.2 MB) drive or a 3.5-inch (1.44 MB) drive may be used.
- Drive B: may be either a low-density or a high-density flexible disk drive. Either a 5.25-inch (360 KB or 1.2 MB) drive or a 3.5-inch (1.44 MB) drive may be used.

Supported Measurement Coprocessor Hardware Configurations

In order to support the measurement coprocessor software, you will need at least 1 MB of RAM on the measurement coprocessor. You need not worry about RAM if you have either of the current products. Both the HP 82300C Measurement Coprocessor and the HP 82324A High-Performance Measurement Coprocessor are shipped with a minimum of 1 MB of RAM. However, if you are using one of the earlier versions of the HP BASIC Language Processor, it may not have enough RAM. The HP 82300A/B BASIC Language Processor was shipped with only 512 KB of RAM as standard equipment. However, you can upgrade it to 1 MB by adding an HP 82303A RAM Expansion Kit. The HP 82300C BASIC Language Processor was shipped with 1 MB of RAM as standard equipment.

The measurement coprocessor software includes an MS-DOS device driver program capable of addressing up to three measurement coprocessors. The three coprocessors — designated "Card 1," "Card 2," and "Card 3" — may be HP 82300C Measurement Coprocessors, HP 82324A High-Performance Measurement Coprocessors, or a combination of both. The standard address and interrupt level configurations for these coprocessors are as follows:

- Card 1: Address 250h (hexadecimal), IRQ7.
- Card 2: Address 280h, IRQ9.
- Card 3: Address 330h, IRQ5.

Both the HP 82300C Measurement Coprocessor and the HP 82324A High-Performance Measurement Coprocessor are set at the factory with the appropriate hardware configuration for "Card 1":

- The HP 82300C Measurement Coprocessor hardware configuration switches are set at the factory to address 250h and interrupt level IRQ7.
- The HP 82324A High-Performance Measurement Coprocessor is also set at the factory to address 250h, but its interrupt level is autoconfiguring. On finding the coprocessor at address 250h, the driver program assigns the interrupt level IRQ7.

If you are using only one measurement coprocessor, it is recommended that you leave it in the factory default configuration. However, you can set the coprocessor to one of the other supported "card" configurations if necessary. For example, you can have one card configured as "Card 2" or "Card 3" (with no "Card 1") to avoid an address conflict. Refer to the hardware installation guide that came with your measurement coprocessor for instructions.

If you are installing more than one measurement coprocessor, you must set each one to a different address. You should try the standard "card" configurations first. If you need to use a non-standard address or interrupt configuration, refer to chapter 11, "Multiple Coprocessors and Special Configurations."

Dealing with MS-DOS

If you have used other MS-DOS applications with your computer, you'll have no difficulty running HP BASIC in the MS-DOS environment. However, you will need to understand a little about MS-DOS commands, and about the hierarchical directory structure used for MS-DOS mass storage. Your best source of information about the MS-DOS operating system is the documentation supplied with MS-DOS.

To save you some time, let's look at a few MS-DOS concepts and commands that you will need to use to install and run HP BASIC.

The MS-DOS Command Line

When you first turn on your computer, after it finishes booting, it will display an MS-DOS prompt. If you have installed MS-DOS on hard disk drive C:, the MS-DOS prompt will normally be:

```
C>
```

This indicates that the *current drive* is drive C: (the hard disk).

The line following the prompt is the MS-DOS *command line*. If you are using PAM (Personal Application Manager) on an HP Vectra PC, the command line is found at the top of the PAM screen. If you are not using PAM, the command line will be the last line displayed. In either case, the *cursor* will appear on the command line after the MS-DOS prompt. You can type an MS-DOS command on this line and execute the command by pressing (Enter). For example, you can use the MS-DOS "COPY" command to copy a file named "FILE1" as follows:

```
COPY C:\FILE1 C:\FILE2  (Enter)
```

Executable Files and Batch Files

You can also execute an MS-DOS *executable* file (one with the extension ".EXE" on its name) from the command line. For example, HP BASIC is started by invoking the executable file "BASIC.EXE" as follows:

```
BASIC  (Enter)
```

If you have created an MS-DOS *batch* file (one with the name extension ".BAT"), you can invoke it from the command line. For example, to invoke the batch file named "BATCH.BAT", type:

```
BATCH (Enter)
```

from the MS-DOS command line.

Modifying the MS-DOS Prompt

The prompt "C>" or "A>" appears in the command line at the top of the PAM screen. If you execute the "DOS COMMANDS" field from PAM, or if you are not using PAM, the same prompt appears in the MS-DOS command line. This prompt does not indicate the current directory — only the current drive.

You can change this prompt by including the command "PROMPT PG" in your "AUTOEXEC.BAT" file. This command changes the prompt for the MS-DOS command line (but not the PAM command line) to include the current directory. Here are some typical prompts:

`C:\>`	*(The current directory is the root directory of drive C:)*
`C:\DOS>`	*(The current directory is the subdirectory \DOS)*

The MS-DOS Directory Structure

MS-DOS uses a hierarchical directory structure for mass storage. In this structure, a drive or disk *volume* is divided up into directories and subdirectories. A directory is specified by its *path name*. Let's look at a few examples.

The top-most directory in the hierarchy is the *root directory*. For drive C:, the path of the root directory is "C:\". The directory with the path "C:\PROJECTS" is a subdirectory under the root directory. "PROJECTS" also can have one or more subdirectories under it, for example: "C:\PROJECTS\SUB-DIR1" and "C:\PROJECTS\SUB-DIR2".

Within a directory, you can specify individual files by using a path name. For example, the path: "C:\PROJECTS\DATAFILE" specifies the file named "DATAFILE" in the directory "C:\PROJECTS".

The MS-DOS file structure is covered in more detail in chapter 7.

Changing the Current Drive

Changing the current drive is easy. If the current drive is C: and you want to change to drive A:, just put a flexible disk in drive A: and type:

`A:` (Enter)

The prompt will change from "C>" to "A>".

Creating a Directory

You can create a subdirectory using the MS-DOS "MKDIR" command. For example, the command:

`MKDIR C:\DOS` (Enter)

creates the subdirectory "C:\DOS" under the root directory of drive C:.

Removing Files and Directories

You can remove files with the MS-DOS "DEL" (delete) command. For example, to remove the file named "JUNK.BAK" from the directory "C:\DOS", you could use the command:

`DEL C:\DOS\JUNK.BAK` (Enter)

You can remove a directory, provided all files have been deleted from it, with the RMDIR command. For example, to remove the directory "C:\SUB", use the command:

`RMDIR C:\SUB` (Enter)

Changing the Current Directory

You can change directories within the hierarchy for any drive with the MS-DOS "CD" (change directory) command. Here are some examples:

`CD C:\`	*Changes the current directory to the root of drive C:*
`CD \`	*Also changes the current directory to the root directory.*
`CD C:\DOS`	*Changes the current directory to "C:\DOS".*

The CONFIG.SYS and AUTOEXEC.BAT Files

You can change your MS-DOS configuration by putting two files into the root directory of your MS-DOS boot disk: "CONFIG.SYS" and "AUTOEXEC.BAT". MS-DOS looks for these files and executes the commands in them each time you reboot the PC. An extensive discussion of what these files do is beyond the scope of this manual. Refer to your *MS-DOS User's Reference* manual for further information. However, you should be aware of the fact that these files affect your configuration, and that certain changes to these files are necessary when installing HP BASIC.

2

Installing the HP BASIC Software

The measurement coprocessor software is provided both on 5.25-inch, double-sided/double-density flexible disks (numbered 1 through 5) and 3.5-inch, high-density flexible disks (numbered 1 and 2). You can use either set of disks to install your HP BASIC system. Use the size appropriate for your flexible disk drive.

The user-interactive "INSTALL" program makes it easy to install the software. However, there are two things that you need to do before you start. First you should make sure that your personal computer is set up correctly, that MS-DOS is installed, and that your computer meets the hardware requirements given in chapter 1. Second, you should install your measurement coprocessor in your computer, following the instructions provided with the coprocessor.

The INSTALL Program

The "INSTALL" program is provided on "Measurement Coprocessor SW" disk 1. This program will install a complete HP BASIC language system on a hard disk or a high-density (1.2 MB, 5.25-inch or 1.44 MB, 3.5-inch) flexible disk. The installation process is automatic — you need only follow the instructions given on the screen. The procedure is the same for either the HP 82300C Measurement Coprocessor or the HP 82324A High-Performance Measurement Coprocessor.

In general, the "INSTALL" program does the following:

1. Install a device driver program named "HPBLP.SYS" in the root directory of your hard disk. (For more information about this driver, refer to chapter 4.)

2. Modify the "CONFIG.SYS" and "AUTOEXEC.BAT" files and reboot the PC.
3. Create an HP BASIC system directory (normally "C:\BLP") if it doesn't already exist and copy all of the necessary files to that directory. (You will be prompted to change disks if necessary.)
4. Boot the measurement coprocessor and start HP BASIC. An HP BASIC "AUTOST" program then loads all language extension binaries and drivers into the measurement coprocessor system and stores the system in the system directory ("C:\BLP") as the system file "SYSBA514".
5. Exit HP BASIC to MS-DOS (or PAM).

Note

The "INSTALL" program will modify your "CONFIG.SYS" and "AUTOEXEC.BAT" files. However, before this happens, "INSTALL" copies "CONFIG.SYS" to "CONFIG.INS" and "AUTOEXEC.BAT" to "AUTOEXEC.INS" (in the root directory). This preserves copies of your original "CONFIG.SYS" and "AUTOEXEC.BAT" files in case the installation process is aborted. Refer to "What to Do If the Installation Fails" for further information.

Existing HP BASIC Installations

If this is a "first-time" installation, that is, you have not installed HP BASIC on your system previously, you can skip this section. If you have installed a previous version of the software, the procedure depends on which version you have installed.

- If you have previously installed version C.00 or a later version of the measurement coprocessor software (formerly called the HP BASIC Language Processor software), there are no special concerns. Just install the new software over the old in your "C:\BLP" directory. The new software system will over-write the old, but any data files or program files that you have created will be preserved.

- If you have previously installed version A.00 through A.02 of the HP BASIC Language Processor software in an "HPW" file system (normally in the directory "C:\HPW"), don't install the new software in the same directory. (You'll get an error message.) Follow these steps.

  1. Install the new software in a separate directory (C:\BLP is the default), leaving the old software undisturbed.
  2. *Use the HP BASIC program "CATCOPY" to copy all of your existing programs and data files to the new directory (C:\BLP).* Refer to *Porting Guide for the Measurement Coprocessor* for details.
  3. *Leave the old software and your existing data and program files in the old directory (C:\HPW) until you are sure that you are done with them.* Once you are sure, you can change the MS-DOS file-protection attribute for the directory C:\HPW, and then delete all of the files. The procedure is given in the section "Deleting an HPW Volume" in chapter 7.

Note

The new HP BASIC installation will function correctly for both the HP 82300C Measurement Coprocessor and the HP 82324A High-Performance Measurement Coprocessor.

Installing HP BASIC on a Hard Disk

You can use the "INSTALL" program to install the measurement coprocessor software on any internal hard disk. Normally, you would install the software on drive C:. The installation procedure is automatic. It requires little input from you. You need only follow the directions given on the screen by the program. The program will prompt you to specify a directory in which to install the software and then will make the necessary modifications to "CONFIG.SYS" and "AUTOEXEC.BAT". (If you prefer to modify "CONFIG.SYS" and "AUTOEXEC.BAT" manually, refer to "Installing HP BASIC Manually" in chapter 11.) The program will then reboot your PC and install the software.

To begin the installation process, insert "Measurement Coprocessor SW" disk 1 (either 5.25-inch or 3.5-inch, whichever is appropriate) in drive A: and type:

A: (Enter)

from the MS-DOS command line. This changes the current drive to "A:". (If you are starting from the PAM command line, you'll have to "press any key" to complete the drive change.)

To start the "INSTALL" program type:

INSTALL (Enter)

The following screen will appear:

```
                    HP BASIC INSTALLATION

You will be prompted for each "Measurement Coprocessor SW" disk
(3.5-inch: Disk 1 or 5.25-inch: Disks 1-4). If not present, a
driver will be installed causing the PC to be rebooted. Simply
follow the directions on the screen to complete the installation.

CAUTION: If the files AUTOST and BLP.MSG exist in the destination
         directory, they will be destroyed. If you want to save
         these files, exit now by typing [CTRL][C].



Press any key to continue
```

Press any key to continue the installation. The program will prompt you for the name of the subdirectory where you want to install HP BASIC:

```
                    HP BASIC INSTALLATION

   Enter the destination drive and directory. Press [Enter]
   to use the default:   C:\BLP
```

Just press Enter to accept the default directory, C:\BLP, or type in your own choice and press Enter. If you have previously installed HP BASIC in C:\BLP, just press Enter to install the new software over the old. That way you'll still have access to all of your programs and data files in the directory.

At this point the program will install the driver "HPBLP.SYS" on your hard disk. It will then modify your "CONFIG.SYS" and "AUTOEXEC.BAT" files. *However, don't worry, the program preserves your original files as "CONFIG.INS" and "AUTOEXEC.INS", respectively.*

The program now must reboot your PC. The following screen appears:

```
                    HP BASIC INSTALLATION


    About to reboot the PC.
    Remove the disk from drive A: so the system can boot.
```

Lift the release lever on drive A: (press the release button for a 3.5-inch disk drive) so the PC can reboot from the hard disk. Once the PC is rebooted, the following screen appears:

```
                    HP BASIC INSTALLATION


    Please insert disk #1 into drive A:, then press [Enter].
```

Re-insert disk 1, then press Enter to continue the installation.

The program will verify the configuration of your system. After several seconds the configuration will be displayed. A typical screen follows.

```
                    HP BASIC INSTALLATION

        Configuration information
        Display subsystem    EGA with 256 KB RAM COLOR
        DOS memory           636 KB
        Disk drives          1 flexible, 1 hard
        NO configuration/compatibility conflicts found

        Press any key to continue
```

Assuming there are no conflicts reported, just press any key to continue with the installation.

Note

If a configuration problem such as an unsupported display type is reported, you may need to abort the installation. Press (Ctrl) (C) to abort the installation and return to MS-DOS. You will then have to follow the procedure described in "What to Do If the Installation Fails" at the end of this chapter.

At this point, the program will proceed automatically. If you are installing the software from 5.25-inch disks, the program will prompt you to insert disks 2, 3, and 4 when appropriate. In each case, insert the requested disk, close the disk drive door, and press (Enter). This is not necessary if you are using 3.5-inch disks — only disk 1 is required. (The "Manual Examples, LIFINIT, and Selected CSUBs" disk is not needed for the installation.)

Once all of the files have been copied to the hard disk, the program will boot the measurement coprocessor. *Don't interrupt this process — the measurement coprocessor must be booted in order to build and store the HP BASIC system file.* When the boot process is completed the following HP BASIC screen will appear:

```
- - - - - - - - - - - - - - - - - - - - - - - - -
|                 BASIC 5.14                     |
| Copyright Hewlett-Packard Company 1981, 1982,  |
|        1983, 1984, 1985, 1987, 1990            |
- - - - - - - - - - - - - - - - - - - - - - - - -
|                  HFS 5.3                       |
| Copyright Hewlett-Packard Co. 1987, 1988       |
| Copyright The Regents of the University        |
|           of California 1979, 1980, 1983       |
| Copyright AT&T 1980, 1984                      |
- - - - - - - - - - - - - - - - - - - - - - - - -
|             RESTRICTED RIGHTS LEGEND           |
| Use, duplication, or disclosure by the U.S.    |
| Government is subject to restrictions as set   |
| forth in subdivision (b)(3)(ii) of the Rights  |
| in Technical Data and Computer Software        |
| clause at 52.227-7013.                         |
|             Hewlett-Packard Company            |
|   3000 Hanover Street, Palo Alto, CA 94304     |
- - - - - - - - - - - - - - - - - - - - - - - - -

Loading Language Extensions and Drivers ...


                                                 User 1    Caps    Running
  EDIT    Continue  RUN    SCRATCH    LOAD ""  LOAD BIN  LIST BIN  RE-STORE
                                                 ""                  ""
```

An HP BASIC "AUTOST" program will load all of the language extension binaries to "build" your HP BASIC system. *Don't interrupt this program.* The program will then store your system on the hard disk as "SYSBA514". Finally, your computer will return to MS-DOS (or PAM). You should receive the message "Installation completed successfully." *Now you can skip the rest of this chapter and go on to "Starting HP BASIC from MS-DOS" in chapter 3.* (If you receive an error message, refer to "What to Do if the Installation Fails" at the end of this chapter.)

Note

There is a possible complication if you have an HP 82300A/B BASIC Language Processor equipped with Series 200 boot ROMs. If the HP BASIC screen appears, but no autostart program runs and no binaries are loaded, the wrong BASIC "SYSTEM" file may have been booted. (You may get the message "BASIC Main 5.0" or "BASIC Main 5.1" instead of "BASIC Main 5.14".) If this happens, refer to "What to Do if the Installation Fails."

Installing HP BASIC on a High-Density Flexible Disk

If you don't have a hard disk drive, you can install an HP BASIC system on a *high-density* flexible disk in drive A:, which must be either a 5.25-inch 1.2 MB or a 3.5-inch 1.44 MB flexible disk drive. You must use drive A: as the "target" drive, and you will need to have a second drive (drive B:) to use as the "source" drive. However, drive B: does not have to be a high-density drive. It can be either a 5.25-inch (360 KB or 1.2 MB) or a 3.5-inch (1.44 MB) disk drive.

The measurement coprocessor software must be installed on an MS-DOS *system disk*. That is the MS-DOS operating system must be installed on that disk before you install HP BASIC.

The procedure is as follows:

- Format a high density (1.2 MB or 1.44 MB) flexible disk in drive A: and install MS-DOS on this disk (if you haven't already done so). Refer to your MS-DOS user documentation for the procedure.
- *Boot the PC with this system disk in drive A:* (turn on the power or press (Ctrl) (Alt) (Del)).
- Insert "Measurement Coprocessor SW" disk 1 (either 5.25-inch or 3.5-inch, whichever is appropriate) in *drive B:* and type:

  `B:` (Enter)

  to change drives to "B:". Now type:

  `INSTALL` (Enter)

- The "INSTALL" program will proceed automatically with the rest of the installation. Just follow the instructions on the screen. When prompted for the directory in which to install BASIC, specify "A:\BLP" or some other directory on drive A:. Refer to the instructions in "Installing HP BASIC on a Hard Disk" for further comments.

Once you have completed the installation successfully, you can go on to "Starting HP BASIC from MS-DOS" in chapter 3. If the installation fails, refer to the following section.

What to Do If the Installation Fails

If the installation fails, the general procedure is to correct the problem that caused the failure and then repeat the installation. The following are some common problems that might cause the installation to fail:

- *Your personal computer doesn't meet some of the hardware requirements outlined in chapter 1.* For example, it might have less than 640 KB of PC RAM. If so, correct the problem and repeat the installation procedure.
- *Your display system is not supported.* For example, CGA display systems are not supported. To correct this, install one of the supported displays listed in chapter 1 and repeat the installation procedure.
- *The measurement coprocessor is not configured to one of the supported address and interrupt configurations:*

  Card 1: address 250h (IRQ7)
  Card 2: address 280h (IRQ9)
  Card 3: address 330h (IRQ5)

  To correct the problem, change the hardware configuration to one of these three choices and then repeat the installation procedure. (You'll need to set both the address and interrupt for the HP 82300C Measurement Coprocessor, but only the address for the HP 82324A High-Performance Measurement Coprocessor. Refer to the hardware installation manual supplied with your measurement coprocessor for the procedure.)
- *You aborted the installation.* If you aborted the installation process at any point (by pressing Ctrl C), you will have to repeat the installation procedure.
- *The wrong BASIC "SYSTEM" file was booted.* This problem may occur if you have an HP 82300A/B BASIC Language Processor equipped with Series 200 boot ROMs. Repeat the installation, but this time press the space bar to interrupt the boot process and then select the correct system file ("SYSBA514") to boot. Refer to "Booting with Series 200 Boot ROMs" in chapter 3 for further information.

- *You have a buffer address conflict with a window manager or expanded memory manager, or with another card (HP 82324A only).* Resolve the conflict as described in "Resolving Buffer Address Conflicts" in chapter 11 and then repeat the installation.

Normally, once you have corrected the problem that caused the installation to fail, you can just start over:

- If you are installing HP BASIC on your hard disk, insert disk 1 in drive A:, type "`A:` (Enter)", and then type "`INSTALL` (Enter)".
- If you are installing HP BASIC on a flexible disk in drive A:, insert disk 1 in drive B:, type "`B:`" (Enter), and then type "`INSTALL` (Enter)".

The installation will proceed normally, except that the PC will not usually need to be rebooted this time.

In rare cases, it may be necessary to start over with your original "CONFIG.SYS" and "AUTOEXEC.BAT" files. To restore your original files, use the MS-DOS "COPY" statement as follows:

```
COPY C:\CONFIG.INS C:\CONFIG.SYS  (Enter)
```

and

```
COPY C:\AUTOEXEC.INS C:\AUTOEXEC.BAT (Enter)
```

Now repeat the installation procedure in the normal manner.

3

Getting Started with HP BASIC

This chapter begins by explaining how to boot the measurement coprocessor and start HP BASIC. Once you have HP BASIC running you'll learn about the HP BASIC programming environment — how to create, edit, store, and retrieve an HP BASIC program.

Starting HP BASIC from MS-DOS

The procedure to boot the measurement coprocessor and start HP BASIC is very simple, especially if you have only one measurement coprocessor. Let's begin by looking at the configuration screen that the device driver "HPBLP.SYS" generates during the PC boot process (when you turn on the power).

The Driver Configuration Screen

You can tell how many measurement coprocessors have been installed, as well as how they have been configured, from the driver configuration screen that appears at the end of the PC boot, just before the MS-DOS prompt appears. (If you are using PAM, the screen appears only momentarily before the PAM screen is displayed.) Let's look at some typical screens.

The following screen indicates that one HP 82324A High-Performance Measurement Coprocessor is present, that it is configured as "Card 1" with the default configuration (address 250h and IRQ7), and that it is ready to boot. (The boot code will be loaded into RAM during the boot process.)

```
Hewlett-Packard Measurement Coprocessor Device Driver Version A.01.00
Copyright (c) Hewlett-Packard Co. 1989, 1990.  All rights reserved.
      Card  Type    Addr   IRQ              Card State
      ----  -----   ----   ---   ----------------------------------
       1    82324   250H    7     Ready to load boot code
       2            280H    9     No measurement coprocessor found
       3            330H    5     No measurement coprocessor found
```

Note

If you have installed more than one measurement coprocessor in your computer, the type of coprocessor and the configuration of each will be given.

The following screen indicates that one HP 82300C Measurement Coprocessor is present, that it is configured as "Card 1" with the default configuration (address 250h and IRQ7), that it is ready to boot, and that it is equipped with PC300 boot ROM version 1.0.

```
Hewlett-Packard Measurement Coprocessor Device Driver Version A.01.00
Copyright (c) Hewlett-Packard Co. 1989, 1990.  All rights reserved.
      Card  Type    Addr   IRQ              Card State
      ----  -----   ----   ---   ----------------------------------
       1    82300   250H    7     Ready to boot (PC300 BOOTROM 1.0)
       2            280H    9     No measurement coprocessor found
       3            330H    5     No measurement coprocessor found
```

Note

If the display says "S200 BOOTROM 3.0" instead of "PC300 BOOTROM 1.0", you have an HP 82300A/B Language Processor equipped with the earlier release boot ROMs. Refer to "Booting with Series 200 Boot ROMs" later in this chapter for special instructions.

Starting HP BASIC for a Single Measurement Coprocessor

Once the PC is booted and the MS-DOS prompt appears, you are ready to start HP BASIC (provided the measurement coprocessor and its software have been installed). In almost all cases you can start BASIC for a single measurement coprocessor by executing the following command from the MS-DOS command line:

```
BASIC (Enter)
```

This command runs the program "BASIC.EXE", which automatically finds the *first* measurement coprocessor present (Card 1, Card 2, or Card 3), checks to see if the coprocessor has been booted, executes the boot program if it hasn't already been booted, and then starts HP BASIC. Thus, if you have only one measurement coprocessor in your system, "BASIC" is usually all you need to type.

Note

It won't matter which directory you are in when you execute the command "BASIC", provided the directory "C:\BLP" (or the directory where you installed HP BASIC) is included in the MS-DOS "PATH" command in the "AUTOEXEC.BAT" file. (The INSTALL program does this automatically for you.)

There are some exceptions to this procedure. Typing "BASIC" may invoke some other BASIC system, such as GW BASIC or BASIC A, if that system is in ROM or comes before HP BASIC in the PATH. There also may be a conflict if your computer contains or has access to more than one HP BASIC system. For further information, refer to "Conflicts in Starting BASIC" later in this section.

When the boot program runs, a "boot screen" is displayed. The following is a typical boot screen for an HP 82300C Measurement Coprocessor. The boot screen for an HP 82324A High-Performance Measurement Coprocessor is similar, but specifies an "MC68030 Processor."

```
Copyright 1990                               :DOS FIXD, C
Hewlett-Packard Company.                        1B SYSBA514
All Rights Reserved.

PC300 BOOTROM 1.0
MC68000 Processor
Keyboard
Bit Mapped Graphics
HP-IB (Meas. Coproc.) at:      7
Serial (COM1) at:              9
Disk drives (LIF) at:         15
DOS port at:                  19
Printer (PRN) at:             26
1048392 Bytes






BOOTING A SYSTEM
Press CTRL-F10 to Exit to DOS
```

The boot program will boot the first HP BASIC system that it finds, which is the system file "SYSBA514" (BASIC 5.14) on hard disk volume ":DOS,C" in this case.

Note

If the boot program can't find a system (it should take no more than a few seconds), you can return to MS-DOS by pressing Ctrl F10. If this happens, refer to "Conflicts in Starting BASIC" later in this section.

When the boot program is completed, the HP BASIC screen appears:

```
 - - - - - - - - - - - - - - - - - - - - - - - - - -
 |                  BASIC 5.14                     |
 | Copyright Hewlett-Packard Company 1981, 1982,   |
 |        1983, 1984, 1985, 1987, 1990             |
 - - - - - - - - - - - - - - - - - - - - - - - - - -
 |                  HFS 5.3                        |
 | Copyright Hewlett-Packard Co. 1987, 1988        |
 | Copyright The Regents of the University         |
 |           of California 1979, 1980, 1983        |
 | Copyright AT&T 1980, 1984                       |
 - - - - - - - - - - - - - - - - - - - - - - - - - -
 |             RESTRICTED RIGHTS LEGEND            |
 | Use, duplication, or disclosure by the U.S.     |
 | Government is subject to restrictions as set    |
 | forth in subdivision (b)(3)(ii) of the Rights   |
 | in Technical Data and Computer Software         |
 | clause at 52.227-7013.                          |
 |             Hewlett-Packard Company             |
 |   3000 Hanover Street, Palo Alto, CA 94304      |
 - - - - - - - - - - - - - - - - - - - - - - - - - -



BASIC Main 5.14                                  User 1    Caps    Idle
  EDIT     Continue RUN      SCRATCH      LOAD ""  LOAD BIN LIST BIN RE-STORE
                                                   ""                ""
```

You can exit BASIC at any time by pressing Ctrl F10, which returns you to the MS-DOS command line. You can then re-enter BASIC by typing:

`BASIC` Enter

again. BASIC starts immediately — the measurement coprocessor has already been booted, so a reboot is not necessary.

Starting HP BASIC for Multiple Measurement Coprocessors

The procedure for starting HP BASIC for multiple measurement coprocessors is similar to that for just one. However, you have to specify *which* measurement coprocessor to boot or restart. Just specify the "card" number after "BASIC" on the MS-DOS command line:

`BASIC 1` (Enter)	*boots (or restarts) Card 1.*
`BASIC 2` (Enter)	*boots (or restarts) Card 2.*
`BASIC 3` (Enter)	*boots (or restarts) Card 3.*

If you specify a non-existent measurement coprocessor, an error results. As usual, if you just type `BASIC` (Enter), the first card present (the lowest numbered) will be booted or restarted.

For further information about multiple coprocessor setups, refer to chapter 11, "Multiple Coprocessors and Special Configurations."

Some BASIC.EXE Boot Options

The "BASIC.EXE" program provides several options, which are described in the following sections. However, you may not need to use any of these options unless you have more than one BASIC system installed. *Once you have booted the measurement coprocessor and HP BASIC is running, you can skip ahead to "Using the HP BASIC Programming Environment," later in this chapter.*

For a complete description of the BASIC.EXE options and syntax, refer to appendix F, "BASIC.EXE Options and the Boot Search Order."

Note

The command "`BASIC`", whether or not it is followed by any options, is always executed from the *MS-DOS* command line (the "MS-DOS prompt").

Once you have booted HP BASIC using any of the boot options that follow, you can exit BASIC to MS-DOS by pressing (Ctrl) (F10). You can then re-enter BASIC (without a reboot) by typing "`BASIC` (Enter)".

Forcing a Boot

If you want to force a reboot of the measurement coprocessor, whether or not it has already been booted, type:

```
BASIC /BOOT  [Enter]
```

from the MS-DOS command line. The measurement coprocessor will be booted with the first system file found. (Note, however, that the measurement coprocessor cannot be booted if it is in background mode.)

You can also specify *which* measurement coprocessor to boot. For example:

```
BASIC 3 /BOOT  [Enter]
```

Booting with a Menu

The command:

```
BASIC /BOOT /MENU  [Enter]
```

(from the MS-DOS command line) forces a reboot, and causes the boot program to display all available systems, letting you choose the one you want to boot. (This option does not work for the old Series 200 boot ROMs.)

Note

You can also obtain a menu of bootable systems by executing the "BASIC /BOOT" option and then immediately pressing the space bar. The boot program will search for and display all bootable systems.

The "menu" of selections appears in the upper right hand corner of the screen. For example, let's consider the following boot screen.

```
Copyright 1990                              :REMOTE, 21, 0, 8
Hewlett-Packard Company.                      1B SYSTEM_BA5
All Rights Reserved.                          2B SYSBA514
                                            :DOS FIXD, C
PC300 BOOTROM 1.0                             3B SYSBA514
MC68000 Processor
Keyboard
Bit Mapped Graphics
HP-IB (Meas. Coproc.) at:       7
Serial (COM1) at:               9
Disk drives (LIF) at:          15
DOS port at:                   19
Printer (PRN) at:              26
1048392 Bytes




SEARCHING FOR A SYSTEM (Select a system or press ENTER to pause)
Press CTRL-F10 to Exit to DOS
```

The boot screen gives you a selection of three systems:

```
:REMOTE, 21, 0, 8
   1B SYSTEM_BA5
   2B SYSBA514
:DOS FIXD, C
   3B SYSBA514
```

The first selection is the system file "SYSTEM_BA5" on the ":REMOTE" volume — an SRM (Shared Resource Management) network. You would type "1B" to select this system. However the file "SYSTEM_BA5" is an earlier version of BASIC, not compatible with "BASIC.EXE", so you would not want to choose "1B".

The second selection is the system file "SYSBA514", also on the SRM network. This a BASIC 5.14 system file, which you can boot by typing "2B". (You don't need to press Enter.)

The third selection is the system file "SYSBA514" on hard (fixed) disk volume ":DOS,C". This is also a BASIC 5.14 system file, which you can boot by typing "3B".

Again, you can specify which measurement coprocessor to boot as follows:

```
BASIC 2 /BOOT /MENU  Enter
```

Booting a Specific System File

A third very useful option provided by BASIC.EXE is that you can specify the file name of the system that you want to boot. For example:

```
BASIC /BOOT SYSBA514  Enter
```

(from the MS-DOS command line) specifies that you want to boot the measurement coprocessor with the system file named "SYSBA514" *in your current MS-DOS directory.* You can also specify the full path name of a system file, for example:

```
BASIC /BOOT \BLP\SYSBA514:DOS,C  Enter

BASIC /BOOT SYSBA514:REMOTE,21,0,8  Enter

BASIC 3 /BOOT \BLP\SYSBA514:DOS,C  Enter
```

Conflicts in Starting BASIC

If your PC has access to or contains more than one BASIC system, you may encounter a conflict when you try to boot or restart HP BASIC. The problem occurs when you type "BASIC" and the computer can't determine the answer to the question "which BASIC?" Let's consider some specific situations.

Conflicts with Earlier Versions of HP BASIC

If you have an earlier version of HP BASIC (language processor software version A.00 through A.02) installed in the directory "C:\HPW", you will need to remove that directory from the "PATH" command in "AUTOEXEC.BAT".

The reason is simple. When you ran the INSTALL program, it appended "C:\BLP" (or whatever directory you installed the new HP BASIC system in) to the "PATH" command. However, if "C:\HPW" was already in the "PATH", it precedes "C:\BLP":

```
PATH=C:\;C:\DOS;C:\HPW;C:\BLP
```

Removing "C:\HPW" from the path will prevent the command "BASIC" from starting the old BASIC system in that directory when you wanted the new system in "C:\BLP". The path becomes:

```
PATH=C:\;C:\DOS;C:\BLP
```

Now, the command "BASIC" will start the new HP BASIC system ("SYSBA514") from any directory except "C:\HPW".

If you want to run the old HP BASIC system, execute the following from MS-DOS:

```
CD C:\HPW  (Enter)
BOOT  (Enter)
BASIC  (Enter)
```

The reason the "BOOT" statement is necessary is that, if you have already booted the measurement coprocessor for the new HP BASIC system, the old HP BASIC system won't run. The "BOOT" statement reboots the measurement coprocessor using the old boot program.

Likewise, once you have booted and run the old HP BASIC system, you will have to reboot the measurement coprocessor in order to start the new HP BASIC system. For example, you could execute the following statements:

```
CD C:\BLP  (Enter)
BASIC /BOOT
```

from MS-DOS to force a boot and start HP BASIC.

Note Earlier versions of the HP BASIC software were available in ROM (the HP 82304 BASIC ROM Board). If you have a ROM system and attempt to boot it with the current-version BASIC.EXE program (the one you installed in C:\BLP), the system will boot, but it will not run correctly.

Conflicts with Other BASIC Languages

Besides HP BASIC, there are other BASIC language systems that use "BASIC" as the run command. Thus, conflicts are possible. If you have another BASIC language installed on your hard disk, you may need to modify the "PATH" command in the "AUTOEXEC.BAT" file to cause HP BASIC to be found first. An alternative is to simply change directories to the one where HP BASIC is found before starting BASIC. For example:

```
CD C:\BLP  (Enter)
BASIC  (Enter)
```

Note This may be the only alternative if you have some other BASIC language in ROM. IBM PC-AT computers and AT-compatible computers may have IBM BASIC A or Microsoft GW BASIC in ROM. If your computer has such a BASIC in ROM, and uses "BASIC" as the run command, you will have to change directories as shown before starting HP BASIC.

Booting HP BASIC from an SRM Network

You can boot an HP BASIC system from an SRM network by using the "/BOOT /MENU" option, as described earlier. On the other hand, you can boot from SRM by simply typing:

```
BASIC /BOOT  (Enter)
```

because the ":REMOTE" volume is searched first. Thus, the system file on the SRM network will be found first, and will be booted. (Refer to appendix F for further information about the boot search order.) You can also specify the specific path to the system file on the SRM. For example:

```
BASIC /BOOT SYSBA514:REMOTE,21,0,8  (Enter)
```

Refer to appendix A, "Using SRM with the Measurement Coprocessor," for further information.

Booting with Series 200 Boot ROMs

The HP 82300A/B BASIC Language Processor came equipped with Series 200 Version 3.0 boot ROMs. The HP 82300C Measurement Coprocessor (formerly called the HP 82300C BASIC Language Processor) comes equipped with PC300 Version 1.0 boot ROMs, which provide a considerably faster boot process.

If you have an HP 82300A/B board, and you haven't upgraded the boot ROMs, you can boot HP BASIC with the old boot ROMs. However, the procedure is slightly different. To determine whether you have Series 200 boot ROMs, reboot your PC. If the screen at the end of the boot process indicates "S200 BOOTROM 3.0" rather than "PC300 BOOTROM 1.0", you have Series 200 boot ROMs. An example of this screen is shown below.

```
Hewlett-Packard Measurement Coprocessor Device Driver Version A.01.00
Copyright (c) Hewlett-Packard Co. 1989, 1990.  All rights reserved.
      Card  Type    Addr   IRQ              Card State
      ----  -----   ----   ---  -----------------------------------
       1    82300   250H    7    Ready to boot (S200  BOOTROM 3.0)
       2            280H    9    No measurement coprocessor found
       3            330H    5    No measurement coprocessor found
```

If you have Series 200 boot ROMs, you will have to specify the name of the system file to boot. For example, execute:

```
BASIC /BOOT SYSBA514  (Enter)
```

from the MS-DOS command line. The boot process will proceed normally, except that a Series 200 style boot screen will appear, and the boot process will take quite a bit longer. The following is a typical Series 200 style boot screen:

```
PC 300                                   :HP9122 FIXD, 1503, 0, 0
Copyright 1982,                             1B SYSBA514
Hewlett-Packard Company.
All Rights Reserved.

BOOTROM 3.0
Keyboard
Graphics
HP-IB
HP98626 at 9
HP98624 at 15
HP98622 at 19
HP98622 at 26
1048416 Bytes
```

If you have Series 200 boot ROMs and you have an HP 82304 BASIC ROM Board, an SRM card, or an HP-IB disk drive installed, there is an additional complication. The Series 200 boot program will find the ROM-based BASIC system, or any BASIC system file on the SRM or HP-IB disk drive, first. The program will attempt to boot that file. To avoid this, type:

`BASIC /BOOT SYSBA514` (Enter)

from the MS-DOS command line. Wait until the "Keyboard" message appears on the boot screen, and then press the spacebar. (The "BASIC /BOOT /MENU" option will not work with the Series 200 boot ROMs.) This will cause the boot program to list the full selection of systems to boot. You can then choose the appropriate system file ("SYSBA514").

For further information about the Series 200 boot search order, refer to appendix F, "BASIC.EXE Options and the Boot Search Order."

Starting HP BASIC with a Batch File

You can use an MS-DOS batch file to start HP BASIC. If you are using any of the boot options, you can avoid retyping the options each time by using the batch file. Batch files are distinguished by appending the extension ".BAT" to the file name (for example, "BLP.BAT").

You can use any convenient MS-DOS editor program capable of storing an MS-DOS ASCII file to create a batch file. For example, you could use Executive MemoMaker or Microsoft Word — just make sure that you store the file as an *unformatted* MS-DOS ASCII file. Refer to the user's manual that came with your editor for instructions. If you do not have such an editor, you can use EDLIN to edit your files. Refer to your MS-DOS documentation for instructions on using EDLIN.

For example, you could create a batch file that consists of the following lines:

```
CD C:\BLP
BASIC /BOOT SYSBA514
```

If you store this file with the name "BLP.BAT", you can boot BASIC by just typing "BLP (Enter)" from the MS-DOS command line.

Autostarting HP BASIC

You can start HP BASIC automatically when the PC is rebooted by including commands at the end of the "AUTOEXEC.BAT" file. (Use your MS-DOS editor to modify "AUTOEXEC.BAT", which is found in the root directory.)

For example, the following "AUTOEXEC.BAT" file will start HP BASIC automatically when the PC is turned on or rebooted:

```
set BLPDIR1=C:\BLP
PATH=C:\;C:\DOS;C:\BLP
CD C:\BLP
BASIC /BOOT SYSBA514
```

If you want an HP BASIC application program to autostart, just write an HP BASIC "AUTOST" program. To do this, write a program that does something that you want to occur whenever BASIC is booted. For example, you could write a program to LOAD and RUN another program. The following

"AUTOST" program will load and run the program named "APP_PROG" whenever HP BASIC is rebooted:

```
10   ! AUTOST program (runs automatically on BASIC reboot).
20   LOAD "\BLP\APP_PROG:DOS,C",1  ! Load and run "APP_PROG"
30   END
```

There are two catches. First, the "AUTOST" program must be a "PROG"-type file, so you must use the STORE statement to write it to disk. Second, the program must be stored under the name "AUTOST" in the directory from which you are going to boot BASIC.

When HP BASIC is booted, the current mass storage is the directory from which you booted. Since the above "AUTOEXEC.BAT" file changes the directory to "C:\BLP" and then boots, you would want to STORE the "AUTOST" program in the "BLP" directory. In other words, write the program, press "Pause" to leave EDIT mode, and then execute:

```
STORE "\BLP\AUTOST:DOS,C"  Enter
```

from the *HP BASIC* command line. (Refer to "Using the Program Editor," later in this chapter, for further information.)

Thus, by using both an MS-DOS "AUTOEXEC.BAT" file and an HP BASIC "AUTOST" program, you can automatically start an HP BASIC application program at power-up.

Starting HP BASIC From PAM

You can add a field to your PAM (Personal Application Manager) screen to start HP BASIC. The procedure is as follows:

1. Press `Manage Applics` (F5) in the PAM menu.
2. Press `Add` (F1).
3. Press `Add Unlisted` (F5).
4. You will be prompted for a path. For example, type:

   ```
   C:\BLP  Enter
   ```

5. You are prompted for a title. For example, type:

   `HP BASIC 5.14` (Enter)

6. You are prompted for a run command. For example, type:

   `BASIC /BOOT SYSBA514` (Enter)

7. Press `Save` ((F1)) to save the configuration.

8. Press `Exit` ((F8)) three times to return to the main PAM screen.

From now on, whenever you highlight the PAM field "HP BASIC 5.14" and press `Start Applic` ((F1)), the run command will be executed.

Using the HP BASIC Programming Environment

This section describes the HP BASIC programming environment, and tells how to create, edit, and run HP BASIC programs.

Some HP BASIC Terms

Let's begin by defining some of the more frequently used HP BASIC terms.

Keyword

A keyword is a group of uppercase characters that is understood by the BASIC language system to represent some predefined action.

Statement

A statement is a keyword (sometimes optional) followed by any parameters, lists, specifiers, and secondary keywords that are allowed with that keyword.

Program Line

A program line contains at least a line number followed by a statement. It may also contain a line label, a name that is placed after the line number and terminated by a colon.

Program

A list of program lines, with an END statement on the last line.

Command

A command is a statement that is typed without a line number and executed. There are some commands that cannot be stored as program lines, such as DEL and SCRATCH. There are also some statements that cannot be executed as commands, such as DIM and RETURN.

Enter

Entering a program line means that you type a line number followed by a valid statement and then press the (Enter) key. The line is stored in memory as part of a program, but it performs no function until you run the program.

Execute

Execute means that you type a statement with no line number and press (Enter). The command is executed immediately and is not stored in a program.

Menu

A "menu" is a set of selections consisting of the softkey labels for keys (F1) through (F8). HP BASIC provides four menus: the "System" menu and the "User 1," "User 2," and "User 3" menus.

The HP BASIC Display Screen

Once you have booted the measurement coprocessor and started HP BASIC, the following HP BASIC screen appears. (The copyright notice in the output area appears only after a reboot.)

```
- - - - - - - - - - - - - - - - - - - - - - - - -
|                 BASIC 5.14                      |
| Copyright Hewlett-Packard Company 1981, 1982,   |
|        1983, 1984, 1985, 1987, 1990             |
- - - - - - - - - - - - - - - - - - - - - - - - -
|                 HFS 5.3                         |
| Copyright Hewlett-Packard Co. 1987, 1988        |
| Copyright The Regents of the University         |
|           of California 1979, 1980, 1983        |
| Copyright AT&T 1980, 1984                       |
- - - - - - - - - - - - - - - - - - - - - - - - -
|            RESTRICTED RIGHTS LEGEND             |
| Use, duplication, or disclosure by the U.S.     |
| Government is subject to restrictions as set    |
| forth in subdivision (b)(3)(ii) of the Rights   |
| in Technical Data and Computer Software         |
| clause at 52.227-7013.                          |
|            Hewlett-Packard Company              |
|  3000 Hanover Street, Palo Alto, CA 94304       |
- - - - - - - - - - - - - - - - - - - - - - - - -


BASIC Main 5.14                                   User 1    Caps    Idle
  EDIT     Continue  RUN     SCRATCH     LOAD ""  LOAD BIN  LIST BIN  RE-STORE
                                                   ""                  ""
```

Note The rest of this chapter discusses HP BASIC function keys and commands. You must enter HP BASIC first, as described in the previous section, before you can execute any of these functions or commands.

The following figure identifies the various areas of the HP BASIC screen:

```
- - - - - - - - - - - - - - - - - - - - - - - - -    OUTPUT/PRINT Area
|                BASIC 5.14                      |
| Copyright Hewlett-Packard Company 1981, 1982,  |
|        1983, 1984, 1985, 1987, 1990            |
- - - - - - - - - - - - - - - - - - - - - - - - -
|                 HFS 5.3                        |
| Copyright Hewlett-Packard Co. 1987, 1988       |
| Copyright The Regents of the University        |
|           of California 1979, 1980, 1983       |
| Copyright AT&T 1980, 1984                      |
- - - - - - - - - - - - - - - - - - - - - - - - -
|          RESTRICTED RIGHTS LEGEND              |
| Use, duplication, or disclosure by the U.S.    |
| Government is subject to restrictions as set   |
| forth in subdivision (b)(3)(ii) of the Rights  |
| in Technical Data and Computer Software        |
| clause at 52.227-7013.                         |
|          Hewlett-Packard Company               |
|   3000 Hanover Street, Palo Alto, CA 94304     |
- - - - - - - - - - - - - - - - - - - - - - - - -
                                                     Blank Line
                                                     DISP Line
LOAD "MYFILE"                                        Keyboard Area (two lines)

BASIC Main 5.14                                      User 1    Caps    Idle
  EDIT    Continue  RUN    SCRATCH    LOAD ""   LOAD BIN  LIST BIN  RE-STORE
                                                  ""                  ""
```

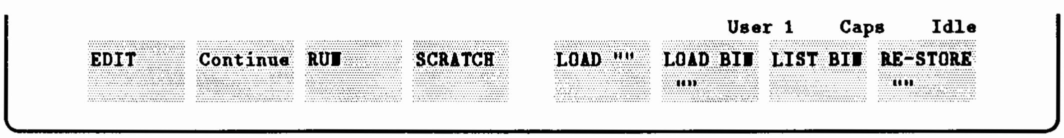
Each area of the screen is explained in the following table:

Label	Description
OUTPUT/PRINT Area	The largest portion of the display. Characters specified in PRINT and OUTPUT statements are displayed here when PRINTER IS CRT is active. System messages are also displayed here when PRINTALL ON and PRINTER IS CRT are *both* active.
DISP Line	Destination of characters specified in DISP statements.
Keyboard Input	Characters (commands) typed on the keyboard appear in this line.
System Message Line	System messages and system status appear in the line just above the softkey labels. The "menu indicator" shows which softkeys are active ("User 1", "User 2", "User 3", or "System"). The "caps lock" indicator ("Caps") toggles on and off with the Caps Lock key. The "program status" indicator shows the status of the currently loaded program ("Running", "Paused", or "Idle").
Softkey labels	Labels of softkeys appear here if KEY LABELS is active.

Using the Keyboard and Softkeys

The measurement coprocessor maps certain keys on your PC keyboard to perform HP BASIC functions. Keyboard overlays are provided to show you this mapping. Refer to chapter 5, "Using the Keyboard with HP BASIC," for further information.

When HP BASIC is first booted, the User 1 softkeys are displayed:



The LOAD (F5), LOAD BIN (F6), LIST BIN (F7), and RE-STORE (F8) softkeys are all *typing aid* keys. That is, pressing one of these keys simply puts the corresponding command in the keyboard area. You have to complete the statement and press Enter to execute it. LOAD and RE-STORE are used to load and store programs. We'll look at this more closely later. LOAD BIN and LIST BIN are used to load and list binaries (language extension programs).

The SCRATCH key (F4) is also a typing aid key. Press SCRATCH Enter to clear a program from memory.

The Continue (F2) and RUN (F3) keys are not typing aid keys. They execute a command directly when pressed. RUN causes the program currently in memory to run, while Continue is used to continue a program that has been paused.

The Edit key (F1) is a typing aid key used to enter the EDIT mode, which is our next topic.

Collectively, the eight softkeys are called a *menu*. In addition to the User 1 menu, BASIC provides the User 2, User 3, and System menus. To move from one menu to another, use the "Menu" (F9) and "System" (F10) keys:

- Press "System" (F10) to move to the System menu. Press Shift F10 to move back to the current User menu. (Press Ctrl F10 to exit BASIC.)
- Press "Menu" (F9) to toggle the softkey labels on and off. Press Shift F9 to cycle through the User menus: User 1, then User 2, then User 3, and then back to User 1.

Using the Program Editor

When you enter EDIT mode, the HP BASIC program editor provides many useful features for creating and modifying programs. You can enter EDIT mode by typing:

EDIT *line number* , *increment* Enter

The *line number* and *increment* parameters are optional. If you just type:

EDIT Enter

the default starting line number is 10, and each line will be incremented by 10. Remember that you can use the EDIT softkey as a typing aid. Pressing EDIT followed by Enter will initiate EDIT mode.

In EDIT mode the format of the screen display is changed, as shown in the following figure.

```
110  A=.03                                   Previous Program Lines (if any)
120  B=.02
130  X=0
140  Y=0
150  C=A+B
160  PRINT " Item     Total    Total"
170  PRINT " Price     Tax     Cost"
180  PRINT "-----------------------"


190  P=0                                     Current Program Line (2 CRT lines)


 _                                           System Message Line (if needed)


200  INPUT "Input item price",P              Following Program Lines (if any)
210  D=P*C
220  E=P+D
230  X=X+D
240  Y=Y+E
250  DISP "Tax =";D;"Item cost =";E
260  PRINT P,X,Y
270  GOTO 190
280  END
                                                          User 2     Caps     Idle
 RENumber  Continue   RUN    MOVELINE    COPYLINE  FIND ""  CHANGE "  INDENT
                             S , TO      S , TO             " TO ""
```

You can view several lines before and after the line you are editing. The system supplies the line number for the current line, and program portions can be viewed by scrolling with the cursor-control (arrow) keys.

The softkey labels also change in EDIT mode. The User 2 softkeys are displayed, as shown above. The `Continue` and `RUN` keys have their usual functions. The other softkeys are all typing aids that you can use to execute editing functions. These functions will be discussed in the following sections.

Entering Program Lines

You enter program lines by typing them after the line number and pressing (Enter). The computer checks for syntax errors and converts letter case to the required form for names and keywords, and then stores the line. The computer supplies a line number automatically. If you want to change the line number, simply back up the cursor to the appropriate position and type in the line number you want. Changing the line number causes a copy operation, not a move. The original line still exists.

Inserting Lines

You can insert new lines between existing lines very easily. For example, assume you want to insert some lines between line 90 and line 100. Place line 100 in the current-line position and press (Shift) (Insert) ("INSERT LINE") on the PC keyboard. The program display "opens" and a new line number appears between line 90 and line 100. Type and store the inserted lines in the normal manner. The computer maintains the established increment between line numbers whenever possible. When the normal increment cannot be maintained, an increment of one is used. When there are no line numbers available between the current line and the next line, enough of the program below the current line is renumbered to allow the insert operation to continue.

Deleting and Recalling Lines

You can delete lines one at a time or in blocks. To delete the current line, press (Shift) (Delete) ("DELETE LINE") on the PC keyboard. If you delete a line by mistake, the line can be recovered by pressing `Recall` (System (F8)). To do this, use the following procedure:

1. Position the cursor below the line where you want to insert the deleted line.
2. Press (Shift) (Insert). The program display will "open" and a new line number will appear.
3. Press `Recall`. The deleted line will appear.
4. Press (Enter). A new line number will appear beneath the line just recalled.
5. You can enter new lines at this point or move to another area of the program for other editing.

When no parameters

number and renumbers the entire program

Moving Program Segments

You can move blocks of text with the MOVELINES command. You can use the **MOVELINES** key (F4) as a typing aid. This command moves contiguous program lines from one location to another. For example, if you wish to move the code in a program that is located between lines 100–250 to a new location in the program beginning with line number 1000, execute:

```
MOVELINES 100,250 to 1000
```

Or, you could specify the lines by using labels:

```
MOVELINES label1,label2 TO new_block
```

Note

If you intend to create a subprogram or function by moving a block of code, enter the subprogram header before moving the code. You cannot enter a SUB or DEF FN statement if there are other statements following it.

If the starting line number does not exist, the next line is used. If the ending line number does not exist, the previous line number is used. If a line label doesn't exist, an error occurs and no moving takes place. If an error occurs during the MOVELINE operation (a memory overflow, for example), the move is terminated and the program is left partially modified.

Copying Program Segments

The COPYLINES command performs the same function as MOVELINES, except that it leaves the code in the old location and copies it to the new location as well. You can use the `COPYLINES` key (F5) as a typing aid. This is desirable when you want a section of code that is very similar, but not identical to a section of code you already have. (If it were identical, you would probably put it into a subprogram.) It is often easier to copy code and modify one version than to type two separate, only slightly different versions.

The FIND Command

You can find all the occurrences of a particular string in a program by using the FIND command. You can use the `FIND` key (F6) as a typing aid. When a program line that contains the specified string has been found, the computer places you in EDIT mode automatically. The current line is the line containing the specified string, and the cursor is positioned on the first character of the string. The message "Found 'string value' " is displayed in the system message line. You can then edit the string as you desire. When you press Enter, the computer will continue its search for the string, stopping when it finds another occurrence of the string, when it reaches the end of the program, or when it reaches the last line of the specified range. To cancel a search operation before it is finished, press "Clr I/O". The following examples illustrate the use of the FIND command.

`FIND "STRINGA"`	*Searches for the first occurrence of the string "STRINGA", starting from the current location in the program.*

`FIND "STRINGB" IN 1500`	*Searches the program for the string "STRINGB" beginning at line 1500.*
`FIND "STRINGC" IN 1550, 1700`	*Searches the program for the string "STRINGC" beginning at line 1550 and ending at line 1700.*

You can use line labels instead of line numbers if you wish.

The CHANGE Command

You can replace any string with another string by using the CHANGE command. You can use the `CHANGE` key (F7) as a typing aid. CHANGE is like FIND in that it looks through your program and finds occurrences of the specified string. However it also makes a tentative change that you can confirm by pressing Enter, or deny by pressing `Continue`. If you are positive that you don't need to verify each replacement, appending ";ALL" to the CHANGE command will cause the search-and-replace to be done with no further action on your part.

```
CHANGE "OLD TEXT" TO "NEW TEXT"
```

The computer searches the entire program from the beginning and stops at any point where it finds the string "OLD TEXT". You are then asked the following question: "OLD TEXT" to "NEW TEXT?". Press Enter if you want the change made, or press `Continue` if you do not. In either case the computer will continue the search, repeating the above process whenever it finds the specified string.

```
CHANGE "OLD TEXT" to "NEW TEXT" IN 2600,3000
```

This performs exactly the same function as the previous command, except that the computer will only perform the search from program line 2600 through program line 3000.

```
CHANGE "OLD TEXT" to "NEW TEXT";ALL
```

This performs the same function as the first command, except that no verification on your part is required. The computer automatically makes the requested change.

Getting Out of EDIT Mode

There are many ways to terminate EDIT mode. If you want to return the CRT to its "normal" mode, press "Pause".

Another way to terminate EDIT mode is to proceed with another operation by pressing the appropriate function key. Initiating operations such as LOAD, CAT, LIST, RUN, STEP, or PAUSE will automatically terminate EDIT mode.

Listing a Program

You can display or list all or part of your program by executing a LIST statement. The LIST statement has parameters that allow you to specify both the range of lines to be listed and the device to which the listing should be sent. If LIST is executed without any parameters, the default action is to list the entire program on the system printer. The default system printer after a power-on or SCRATCH A is the CRT. The system printer is defined by the PRINTER IS statement.

You specify starting and ending line numbers in the LIST statement, or you may specify labels instead. For example:

`LIST 100,200`	*Lists lines 100 through 200, inclusive.*
`LIST 1850`	*Lists the program from line 1850 to the end.*
`LIST Rocket`	*Lists the program from the line labeled "Rocket" to the end.*

If you want the listing to be printed on an external printer, you must use the PRINTER IS statement prior to the LIST statement:

```
PRINTER IS 26
```

To make the CRT the system printer again use:

```
PRINTER IS 1
```

You can also use the LIST statement to list a program on the printer and keep the CRT as the system printer:

```
LIST #701
LIST #26
```

This statement sends the entire program listing to an HP-IB printer (address 01) without changing the system printer selection.

Indenting

You can indent your program in appropriate places by using the INDENT command. You can use the **INDENT** key ((F8) in the User 2 menu) as a typing aid. This command automatically indents whenever there is the beginning or end of a program statement which causes looping, is conditionally executed, or is a separate program segment. There are two parameters, starting column number (default = 6) and increment (default = 2). The starting column number is the column in which the first character of the first statement of each context appears. The increment specifies the number of spaces that the beginning of the lines move to the left or right when the nesting level of the program changes.

Indenting a program may cause the length of some of the lines to become longer than the computer can list. This condition is indicated by the presence of an asterisk (*) after the line numbers of affected lines. If this occurs, the program will run properly, store properly, and load properly. However, you cannot do a SAVE, then a GET. Doing an INDENT with smaller values will alleviate this problem.

Let's look at an example to see the effect of the INDENT command. The example program "INDNTPGM" is provided on your "Manual Examples, LIFINIT, and Selected CSUBs" disk. To load this program from disk, insert the disk in drive A: and type:

LOAD "\EXAMPLES\INDNTPGM:DOS,A" (Enter)

Now press EDIT (Enter) to enter EDIT mode.

The program should appear as follows:

```
10 FOR I = 1 TO 5
20 REPEAT
30 INPUT "How old are you?",Age
40 Reasonable = 1
50 IF Age < 0 THEN
60 DISP "Forgive me, but you can't be ";Age;"years old."
70 Reasonable = 0
80 ELSE
90 IF Age >= 120 THEN
100 DISP "That's a little difficult to believe."
110 Reasonable = 0
120 ELSE
130 IF Age >= 100 THEN
140 DISP "You are getting up there, aren't you?"
150 ELSE
160 IF Age >= 60 THEN
170 DISP "I'm impressed.  You don't look that old."
180 ELSE
190 IF Age >= 40 THEN
200 DISP "Ah, you're over the hill."
210 ELSE
220 DISP "So, just a youngster."
230 END IF
240 END IF
250 END IF
260 END IF
270 END IF
280 WAIT 2
290 UNTIL Reasonable
300 DISP "You were";Age*365.242198781;"days old on your last birthday."
310 WAIT 2
320 NEXT I
330 END
```

Now press INDENT (Enter). The program should now be indented as shown on the following page. The indents show each level of branching.

```
10  FOR I = 1 TO 5
20    REPEAT
30      INPUT "How old are you?",AGE
40      REASONABLE = 1
50      IF AGE < 0 THEN
60        DISP "Forgive me, but you can't be ";AGE;"years old."
70        REASONABLE = 0
80      ELSE
90        IF AGE >= 120 THEN
100         DISP "That's a little difficult to believe."
110         REASONABLE = 0
120       ELSE
130         IF AGE >= 100 THEN
140           DISP "You are getting up there, aren't you?"
150         ELSE
160           IF AGE >= 60 THEN
170             DISP "I'm impressed.  You don't look that old."
180           ELSE
190             IF AGE >= 40 THEN
200               DISP "Ah, you're over the hill."
210             ELSE
220               DISP "So, just a youngster."
230             END IF
240           END IF
250         END IF
260       END IF
270     END IF
280     WAIT 2
290   UNTIL REASONABLE
300   DISP "You were";Age*365.242198781;"days old on your last birthday."
310   WAIT 2
320 NEXT I
330 END
```

Running a Program

You run a program by pressing the **RUN** key or by typing RUN and pressing Enter. This tells the computer to go through a pre-run phase and then begin normal program execution with the lowest numbered line in the main program. The RUN command can also be followed by a line identifier that lets you specify where the program execution is to begin.

Program Execution

The process of program execution as implemented by the BASIC interpreter is summarized below.

1. Determine which program line is to be acted upon next.
2. Identify the statement that follows the line number and label (if any) on that line.
3. If the statement has a run-time action, perform that action.
4. Repeat steps 1 through 3 until an END, STOP, or PAUSE statement is executed.

The RUN command determines which line is acted on first. Executing RUN with no parameters, or by pressing the RUN key, causes the execution process to begin at the first line of the program. Execution can be started anywhere in the program by using the RUN command with a line identifier. For example:

```
RUN 200
```

This command causes execution of the program to begin at line 200. If there is no line 200, execution begins with the line whose number is closest to and greater than 200. The line identifier can also be a label. For example:

```
RUN Spot_run
```

This command causes execution of the program beginning with the line labeled "Spot_run". If there is no such label, an error results.

Live Keyboard

The term "live keyboard" is used when talking about commands that are executed from the keyboard while a program is running. The keyboard is still active when a program is running. You can execute commands, change variables, and change the state of the computer.

Pausing and Stopping

If the operator does not intervene, a program will run until it encounters an END, PAUSE, or STOP statement. For example, if you wish to pause program execution before its normal completion, press "Pause". This causes a temporary halt to program execution. To continue, press **Continue**. If you wish to stop the program, press "Stop".

The "Run Light"

You can determine the current state of the computer by the indicator in the lower-right hand corner of the CRT. The character in this corner is referred to as the "run light". The following table defines the various indications of the run light.

Status Indicator	Run Light	Computer State
Idle	blank	Program stopped; continue not allowed
Running		Program running
Paused	–	Program paused; may be continued
Transfer	IO	Program paused, but a TRANSFER is still active
Input	?	Computer is waiting for input from the keyboard
Command	*	Computer is executing a command from the keyboard

An Example

To demonstrate some of the interaction between a program and the keyboard, use the EDIT mode to enter the following program.

```
10 DISP "NEXT COMMAND?"
20 X=0
30 PRINT X;
40 X=X+1
50 WAIT .1
60 GOTO 30
70 END
```

1. After you have entered the program, run the program by pressing **RUN**. This will automatically get you out of EDIT mode and begin running the program.

2. Press “Pause”. The printout of numbers stops, and all data on the CRT remains unchanged. The run light indicates that the program is paused and can be continued. The program line that appears at the bottom of the CRT is the next line of the program that will be executed when program execution resumes.
3. Press `Step` a number of times. The program is now executed one line at a time, as indicated by the program lines changing at the bottom of the screen. Notice that the program is still paused and continuable after each press of the `Step` key.
4. Press `Continue`. The printout on the CRT resumes with the next number in sequence, and the run light indicates the program is running.
5. Press “Stop”. The printout of numbers stops, and all the data on the CRT remains unchanged. The run light is off.
6. Press `Continue`. An error results because a stopped program cannot be continued.
7. Press `RUN`. The program runs again, but the number sequence has started from the beginning. RUN causes the program to start from the beginning, not resume.
8. Type x = 0 and press (Enter). Notice that the numbers being printed start over from “1”. The live keyboard was used to change the value of “X”, and the program used this new value immediately.
9. Type WAIT 5 and press (Enter). Notice that the run light changes to indicate that a keyboard command is being executed. The printout is delayed for five seconds while the keyboard command is processed.
10. Press PAUSE, and then type EDIT 50 and press (Enter). The display on the CRT changes to show the program, and line 50 appears in the current-line position of the screen. The run light indicates that the program is paused.
11. Change line 50 to WAIT 2 and press (Enter). The new line 50 is entered, but the run light goes out. Changing the program caused it to move from the paused state to the stopped state.
12. Press `Continue`. An error results. Once a program has been changed, the program is no longer paused, and the CONT command is not allowed.

Clearing a Program from Memory

You can clear an HP BASIC program from computer memory at any time with the SCRATCH command. Press `SCRATCH` (Enter) to clear memory and then press `EDIT` (Enter). On entering EDIT mode you'll find that memory is clear, so you can create a new program.

You don't have to use SCRATCH to clear memory if you are going to LOAD or GET a program. Both of these commands clear computer memory automatically and then retrieve the program from mass storage.

Storing and Retrieving a Program

The previous sections have shown you how to enter, edit, and run a program. The next logical step is to save the program for future use or further development.

The exact procedure for storing and retrieving programs depends upon the type of mass storage device you are using. Your computer may have an internal flexible disk drive, an internal hard drive, an SRM system, or one of the many external disk drives that are compatible with your system.

What Is Mass Storage?

As the adjective "mass" suggests, mass storage devices are data-storage devices which are generally capable of storing "large" amounts of data. Just how much data constitutes a large amount depends on the device itself. Most mass storage devices are capable of storing hundreds of thousands to several million items.

Besides having the ability to store data, mass storage devices are capable of providing means for keeping data organized so that logical groups may be accessed systematically and efficiently. Data items are organized into logical groups of data known as files; a file is merely a collection of data items. Mass storage directories are composed of one or more files. On some HP mass storage devices, a directory consists of all files on the mass storage media; mass storage media are the actual physical means by which data are stored. For example, the media used by the internal drive of your computer consists of magnetic particles on a plastic disk which can be magnetized to store data.

For a complete explanation of the measurement coprocessor DFS (DOS File System) mass storage system, refer to chapter 7, "File Systems and Mass Storage." For now, let's look at some basic concepts.

Mass Storage Specifiers

Before you can store your program, you must specify just where in mass storage you want to store it. This is even more important when you want to retrieve a program — you have to know where it is.

In HP BASIC you specify mass storage with a Mass Storage Volume Specifier, or MSVS. (The MSVS is sometimes called a Mass Storage Unit Specifier, or MSUS.) In the DFS system, the MSVS of your internal flexible disk drive A: is ":DOS,A". The MSVS ":DOS,C" specifies the *current directory* within MS-DOS hard disk drive C:. You can specify an individual directory in an MSVS. For example, the MSVS "\BLP:DOS,C" specifies the directory "BLP" in volume ":DOS,C" (the MS-DOS directory "C:\BLP").

If you include an MSVS in a MASS STORAGE IS (MSI) statement, you can change the *current mass storage*. The current mass storage is accessed by all mass storage operations unless some other mass storage is specified. For example, the statement:

MASS STORAGE IS "\BLP:DOS,C"

or

MSI "\BLP:DOS,C"

makes the directory "BLP" on volume ":DOS,C" the current mass storage.

For further information about DFS mass storage, refer to chapter 7, "File Systems and Mass Storage." Chapter 7 also describes the LIF and HPW file systems, which are provided for compatibility with other HP BASIC systems.

Recording a Program

To record a program, you can use the SAVE or STORE command with a suitable file name. The command used depends upon the type of file you want. If SAVE is used, the text of the program is recorded in an ASCII file. If STORE is used, the program is recorded in a PROG file. The main advantage of a PROG file is rapid access. The following table gives a brief summary of the differences between SAVE and STORE.

	SAVE	**STORE**
File type created:	ASCII	PROG
Retrieved by:	GET	LOAD
Can file be read as data?	Yes	No
Arbitrary program segments allowed?	Yes	No

To store a program, type the keyword STORE followed by a file name, and press (Enter). For example, the command to create a file called "mortgage" is:

```
STORE "MORTGAGE"
```

Note that, since no MSVS is specified in the STORE command, the program is stored on the current mass storage specified by the last MASS STORAGE IS statement.

The SAVE procedure is similar except that SAVE allows you to use line identifiers to specify what portion of the program you want to save. This is helpful when moving or appending program segments during major editing operations. To save all of a program in a file called "WHALES", execute the following command:

```
SAVE "WHALES"
```

The next command saves the last part of a program, from line 500 to the end, in an ASCII file called "TEMP".

```
SAVE "TEMP", 500
```

When both the starting and ending lines are specified, any portion of a program can be saved. Executing the command

```
SAVE "SORTCODE",Sort,Printout
```

saves that portion of the program that is between the lines labeled "Sort" and "Printout" (inclusive) in an ASCII file called "SORTCODE".

Note You can only use SAVE and STORE when first recording a file. If the file name your are trying to use already exists, you will get an error message. To save or store a program to an existing file name, you must use RE-SAVE or RE-STORE. This gives you some protection against accidentally overwriting a file.

Retrieving a Program

Programs saved in an ASCII file are retrieved with the GET statement. Programs stored in a PROG file are retrieved with a LOAD statement. These statements can be executed from the keyboard as commands or included in a program. To retrieve a program you need to know the name and type of the file in which it is stored. If you are not sure of either of these, use the CAT command. The catalog display shows the name and type of all files on the disk.

Note You don't have to use SCRATCH to clear memory before you LOAD or GET a program. Both of these commands clear computer memory automatically and then retrieve the program from mass storage.

Using GET as a Command

You can use the GET command to bring in programs or program segments from an ASCII file, with the options of appending them to an existing program or beginning program execution at a specified line.

If you want to clear any existing program from memory and bring in the contents of an ASCII file, type:

```
GET "FORMULA"
```

This command clears the computer's memory and brings in the ASCII file called "FORMULA". If the first line of the file is not a valid program line, the GET is not performed and an error 68 is reported. If the file is not an ASCII file, the GET is not performed and an error 58 is reported. If you want to append the contents of an ASCII file to an existing program, a line identifier is added to the GET command. For example, assume you have a program in memory whose last line number is 74, and you want to append the contents of a file called "NUMBERS". You can use the following command to accomplish this:

```
GET "NUMBERS",750
```

This appends the program lines from the file called "NUMBERS" to the existing program, renumbering them to start with line number 750.

If the specified renumbering would create an invalid line number, an error is sent to the system printer with an error message, but it is not entered into program memory.

The GET command can also specify that program execution is to begin. This is done by adding two line identifiers: one specifies the placement and renumbering just described, and the other specifies the line at which execution is to begin. For example, assume there is no program in memory and that an ASCII file called "RATES" contains valid program lines. A typical command to bring a program into memory and begin execution at the first line is:

```
GET "RATES",10,10
```

If there is already a program in memory, an append run is allowed. For example:

```
GET "RATES",250,100
```

This command specifies that any existing lines from 250 to the end are to be deleted, the contents of "RATES" is to be renumbered and appended beginning at line 250, and then program execution is to begin at line 100.

Using GET in a Program Line

The GET statement can be used in a program to transfer execution from one program segment to another. This example of a programmed GET demonstrates a simple linkage of two program segments, as might occur when the entire program is too large to fit in available memory.

First Program Segment:

```
10 COM Ohms,Amps,Volts
20 Ohms = 120
30 Volts = 240
40 Amps = Volts/Ohms
50 GET "WATTAGE"
60 END
```

File WATTAGE:

```
10 COM Ohms,Amps,Volts
20 Watts = Amps*Volts
30 PRINT "Resistor Ohms =";Ohms
40 PRINT "Resistor Wattage =";Watts
50 END
```

The COM statement dimensions and reserves memory for variables in a special common memory area so more than one program can access the variables.

Using LOAD as a Command

The LOAD command is used to bring in programs from a PROG file, with the option of beginning program execution at a specified line. For example:

```
LOAD "CANNON"
```

This command clears memory and loads the contents of the PROG file called "CANNON". If the file is not a PROG file, the LOAD is not performed and an error 58 is reported. If any lines require a language extension that is not currently installed, those lines cannot be executed. However, the LOAD proceeds without error.

The LOAD command can also specify that program execution is to begin.

```
LOAD "STONE",10
```

This command causes the computer to load the program in file "STONE" and begin execution at line 10. The line identifier may be a label or a line number, but it must identify a line in the main program segment, not in a subprogram or user-defined function.

The LOAD command cannot be used to bring in arbitrary program segments or append to a main program like GET can.

Using LOAD in a Program Line

When used in a program line, the actions of the LOAD statement are the same as those described for the LOAD command, except program execution resumes whether a line identifier is specified or not. For example:

```
120 LOAD "PART2"
```

When this program statement is executed, the existing program is replaced by the contents of the PROG file called "PART2", and program execution resumes with the first line in the new program.

Exiting HP BASIC

You can exit HP BASIC at any time, including in EDIT mode or while a program is running, by pressing "Exit" ((Ctrl) (F10)). You will be returned to the MS-DOS command line, or to PAM, depending on where you started.

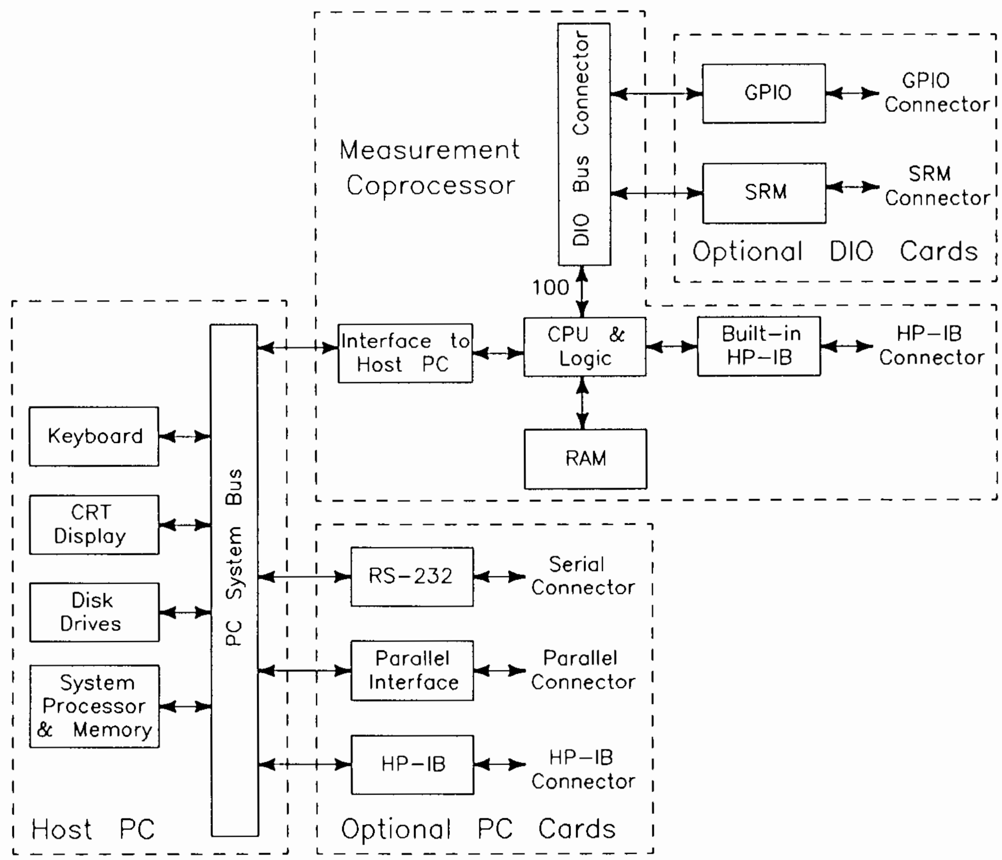
4

An Overview of the Measurement Coprocessor System

Hewlett-Packard offers two measurement coprocessor products: the HP 82300C Measurement Coprocessor and the HP 82324A High-Performance Measurement Coprocessor. Either coprocessor provides HP 9000 Series 300 architecture running HP Series 200/300 BASIC, and turns your HP Vectra PC (or AT-compatible PC) into a multiprocessing computer aided test system. The HP 82300C Measurement Coprocessor has an MC 68000 CPU (central processing unit) running at 8 MHz with from 1.0 to 4.0 MB of RAM. The HP 82324A High-Performance Measurement Coprocessor has an MC 68030 CPU running at 16 MHz with from 1.0 to 8.0 MB of RAM. The HP 82324A may have an optional MC 68882 FPU (floating point unit) installed for increased computational performance. In general, the HP 82324A offers identical functionality to the HP 82300C, but with more than double the speed for most operations.

Each measurement coprocessor occupies a PC slot, and draws its power from the PC I/O bus. The coprocessor accesses such PC resources as the keyboard, display, disk drives, and PC I/O cards through the PC bus. The measurement coprocessor provides a "built-in" HP-IB interface on its back panel. Two DIO "sister" cards are available: the HP 82306A GPIO Interface and the HP 50963A SRM Card. These DIO cards plug into a PC slot, but are connected to the measurement coprocessor with DIO cables.

The following block diagram shows the relationship of the measurement coprocessor (either the HP 82300C or the HP 82324A) and the host PC:



The measurement coprocessor software has several components, including:

- HP 9000 Series 200/300 BASIC 5.14.
- A complete set of language-extension binaries (some of which are customized for the measurement coprocessor).
- An *emulator* program and several driver programs that allow the measurement coprocessor to access standard PC peripheral devices as though they were Series 200/300 devices.

The measurement coprocessor software performs several tasks to emulate an HP BASIC workstation:

- It maps the keys of the PC keyboard to perform HP BASIC functions. (Refer to chapter 5.)
- It emulates an HP 9000 Series 300 "bit-mapped" display using standard PC display adapters and monitors. (Refer to chapter 6.)
- It provides direct HP BASIC access to the PC internal disk drives (both flexible disk and hard disk drives). (Refer to chapter 7.)

In addition, the measurement coprocessor has access to its own internal HP-IB interface, to interface cards installed in PC slots, and to MS-DOS through the MS-DOS communications port. The measurement coprocessor is capable of operating in "foreground" (normal) mode and in "background" mode. In background mode, the measurement coprocessor can continue to operate, running a program, while you have full access to MS-DOS. The rest of this chapter covers these topics.

Available Devices and Interfaces

The measurement coprocessor uses computer devices as though they were standard HP 9000 Series 200/300 devices. HP BASIC accesses most peripherals with standard select codes and device selectors.

Select Codes and the Measurement Coprocessor

The following table shows the select codes used to access several interfaces and devices from the measurement coprocessor.

Select Code	Interface Selected	Comments
7	Built-in HP-IB	The back-panel HP-IB is equivalent to Series 200/300 built-in HP-IB.
9	PC serial port (COM1 or COM2)	Emulates HP 98626A Serial Interface for Series 200/300.
23	PC serial port (COM1 or COM2)	Emulates HP 98626A Serial Interface for Series 200/300.
15	PC disk interface (LIF and HPW only, refer to chapter 7)	Select code 15 emulates an HP-IB disk interface. It is used only to access LIF and HPW volumes on the internal PC disk drives. Select code 15 treats these disk drives as CS80 drives. Note that Select code 15 is not needed for DFS access to the PC disk drives.
19	MS-DOS communications port	The MS-DOS communications port is a direct channel to the MS-DOS operating system.
21*	HP 50963A SRM Card	DIO "sister" card. Equivalent to the HP 50962A SRM Card for Series 200/300.
24	HP 82335A or HP 82990A HP-IB Interface	Emulates HP 98624A HP-IB Interface for Series 200/300.
25	HP 82335A or HP 82990A HP-IB Interface	Emulates HP 98624A HP-IB Interface for Series 200/300.
26	Any system interface (normally a parallel printer interface) at LPT1.	Emulates HP 98622A GPIO Interface used as parallel printer interface.
28*	HP 82306A GPIO Interface	DIO "sister" card. Equivalent to the HP 98622A GPIO Interface for Series 200/300.

* Factory setting. Switches on the card may be changed to other available select codes.

Note

READIO, WRITEIO, CONTROL, STATUS, ENTER, and OUTPUT operations are not supported for select code 15.

You must have the HPIB and CS80 binaries loaded in order to use select code 15. Also, you must have the GPIO binary loaded in order to use select codes 19 or 26. All of these binaries are loaded in the normal installation procedure.

PC Interface Cards

The measurement coprocessor supports several standard PC interface cards plugged into the PC bus, as follows:

- Up to two HP-IB PC interface cards, select codes 24 and 25.
- Up to two serial PC interface cards, select codes 9 and 23.
- One printer interface at select code 26.

HP-IB PC Cards

The measurement coprocessor supports up to two HP 82335A or HP 82990A HP-IB interface cards located in PC slots. One card can be at select code 24, the other at select code 25. The measurement coprocessor software uses these PC cards to emulate the Series 200/300 HP 98624A HP-IB Interface. The Control and Status registers are identical. (These registers are described in the HP BASIC *Language Reference* manual.) All functions of the Series 200/300 HP-IB cards are available except:

- PASS CONTROL is not supported for the HP 82335A or HP 82990A HP-IB cards.
- Direct Memory Access (DMA) is not supported.

In order to use PC HP-IB cards at select codes 24 and 25, you must first configure them as described in chapter 8. The HP 82990A HP-IB card has a default bus address of 30. Note that HP-IB cards from other manufacturers are not supported by the measurement coprocessor.

Serial PC Cards

The measurement coprocessor supports up to two serial ports (COM1 or COM2) at select codes 9 and 23. The measurement coprocessor software uses the PC serial ports to emulate the Series 200/300 HP 98626A Serial Interface. All functions of the Series 200/300 serial card are available except:

- The TRANSFER statement is not supported for select codes 9 and 23.
- Serial card interrupts are not supported.
- Direct Memory Access (DMA) is not supported.

In order to use the serial ports at select codes 9 and 23, you must first configure them as described in chapter 8.

Printer Interface

One printer is supported as defined by LPT1. The GPIO "PSTS" line is used to signify an error condition such as being out of paper. LPT1 emulates a Series 200/300 HP 98622A GPIO card used as a parallel printer interface, although many bits in the registers are not functional.

The printer supported at select code 26 will respond with timeouts just as an HP-IB printer on the internal select code 7, except that the timeout unit when using select code 26 is half-minutes instead of seconds. Thus, `ON TIMEOUT 26, 1` will provide a 30 second timeout, but `ON TIMEOUT 7, 1` will provide a 1 second timeout.

HP-IL printers are not supported using select code 26.

DIO Interface Cards

There are currently two DIO (direct input/output) "sister" cards for the measurement coprocessor. These are the HP 82306A GPIO Interface Card and the HP 50963A SRM Card. Both of these cards have hardware that is equivalent to the corresponding Series 200/300 DIO cards. However, these cards are designed to be installed in a PC slot. The DIO cards take their power from the PC bus, but communicate through a DIO bus connection at the top of the measurement coprocessor. They provide the exact functionality of the corresponding Series 200/300 cards.

Note On a Series 200/300 computer it is possible to send a character and then execute an ENABLE INTR command. This is *not* possible on a PC. Interrupts on the PC should be enabled *before* accessing the PC card.

The HP 82306A GPIO Interface Card

The HP 82306A GPIO Interface Card allows GPIO (General Purpose Input/Output) interfacing to a wide variety of peripherals and instrumentation. This card is equivalent to the HP 98622A GPIO Card for Series 200/300 computers. The Control and Status registers are fully supported, as described in the HP BASIC *Language Reference* manual.

The HP 50963A SRM Card

The HP 50963A SRM Card is used to connect the measurement coprocessor to an SRM (Shared Resource Management) network. This card is equivalent to the HP 50962A SRM Card for Series 200/300 computers. The Control and Status registers are fully supported, as described in the HP BASIC *Language Reference* manual.

The MS-DOS Communications Port

The MS-DOS communications port (select code 19) allows measurement coprocessor programs to communicate with the MS-DOS operating system on your host PC. You can execute special MS-DOS communication port commands to effect changes in the operation of the measurement coprocessor. Or you can execute standard MS-DOS commands from within an HP BASIC program line. Let's look at the communication port control commands first.

Communications Port Control Commands

The MS-DOS communication port commands BACKGROUND, EXIT, BLPSTATUS, SAVE_MODE_ON, SAVE_MODE_OFF, WAIT_ON, and WAIT_OFF are discussed in the following sections. *You can execute these commands by including them in HP BASIC "OUTPUT 19" statements.*

BACKGROUND

The BACKGROUND command allows you to leave an HP BASIC program running in background while you operate MS-DOS software on your computer. You can enter background mode by executing:

```
OUTPUT 19;"BACKGROUND"
```

either from the HP BASIC keyboard entry area, or within a BASIC program. You can also invoke background mode by pressing the "Backgnd" key (Ctrl F9).

Background operation is discussed in more detail in the section "Background Mode and the Measurement Coprocessor" later in this chapter.

EXIT

The EXIT command allows an orderly return from HP BASIC to MS-DOS. To exit HP BASIC type `OUTPUT 19; "EXIT"` and press Enter. Or, just press "Exit" (Ctrl F10).

Caution

To protect file integrity, you must explicitly close all HP BASIC files before exiting HP BASIC. The recommended method of accomplishing this is to stop all HP BASIC programs before executing the EXIT command. Thus, you should not execute `OUTPUT 19; "EXIT"` from within a program.

BLPSTATUS

The BLPSTATUS command provides status information about the measurement coprocessor and your PC system. You can include the following statements in a program:

```
OUTPUT 19;"BLPSTATUS"
ENTER 19;A$,B$,C,D$,E
```

The parameters in the ENTER statement are variables in your program. Upon completion of the above statements, the variables will contain the following information:

- A$ — The version number of the measurement coprocessor software you are using (for example "C.01.00").
- B$ — The viewable size of your display in the format nnnxnnn (horizontal × vertical). For example: "640x480".
- C — The hardware version number: "0" is returned for the HP 82300C Measurement Coprocessor and "1" is returned for the HP 82324A High-Performance Measurement Coprocessor.
- D$ — System type. For example: "PC300".
- E — The x/y (horizontal/vertical) ratio of the physical pixel size on the display currently being used. (1.00 for a VGA display.)

SAVE_MODE_ON

In normal operation, when you access the MS-DOS communications port with an "OUTPUT 19" statement, the HP BASIC alpha screen is saved and is restored on return to HP BASIC. This normal operation is known as "SAVE_MODE_ON" operation. The only problem is that information on the graphics screen is not saved. (This is true whether you are using the combined mode or the separate mode alpha and graphics.)

If you are executing an MS-DOS command (with OUTPUT 19) that does not write to the display (such as DIR > HPW_PIPE), you may retain the graphics information by first executing:

```
OUTPUT 19; "SAVE_MODE_OFF"
```

		the display	

As long as MS-DOS does not write to the screen, SAVE_MODE_OFF allows you to retain graphics information. Otherwise, use SAVE_MODE_ON.

WAIT_ON

After you execute an "OUTPUT 19" statement, the system will wait for a key to be pressed before returning to HP BASIC. If you do not want to wait for a key press, execute:

```
OUTPUT 19; "WAIT_OFF"
```

from HP BASIC first. You can restore the delay by executing:

```
OUTPUT 19; "WAIT_ON"
```

The initial value of this parameter is established in the configuration file. The default configuration file establishes WAIT_ON as the initial value. (Refer to chapter 8 for further information.)

Standard MS-DOS Commands

You can execute standard MS-DOS commands from HP BASIC using an "OUTPUT 19" statement. Any commands other than BACKGROUND, EXIT, BLPSTATUS, SAVE_MODE_ON, SAVE_MODE_OFF, WAIT_ON, or WAIT_OFF will be processed by the MS-DOS command shell, either PAM or COMMAND.COM. (If an MS-DOS error occurs, the MS-DOS error message will appear in the display.)

For example, to execute a DIR command from the keyboard (in HP BASIC), execute:

```
OUTPUT 19;"DIR"
```

The directory will be displayed on the screen. Press any key to restore the HP BASIC display. Note that the directory listed may not be the same as the current directory for HP BASIC. This is because the "current MS-DOS directory" may differ from the "current mass storage" directory for HP BASIC. (Refer to chapter 7 for additional information.)

Using the ENTER Statement

You can use the ENTER statement to obtain results of MS-DOS communications port control commands and MS-DOS commands. You can use ENTER as shown in a program called "ENTERDEM.O", which is found in the directory "EXAMPLES" on your "Manual Examples, LIFINIT, and Selected CSUBs" disk. For example, "ENTERDEM.O" has an `OUTPUT 19; "DIR > HPW_PIPE"` statement followed by an `ENTER 19, B$` statement. If the MS-DOS file "COMMAND.COM" cannot be found, the `ENTER 19;B$` statement will provide the following message:

```
ERROR: return value of -1 for DOS COMMAND "DIR > HPW_PIPE"
```

If an error occurs when the command is being executed, an MS-DOS error message will appear on the screen just as it does when you are running from MS-DOS.

If there are no errors, the next ENTER from select code 19 will get bytes from a file called "HPW_PIPE". Subsequent ENTERs continue to draw bytes from this file.

Executing an ENTER on select code 19 will produce a timeout if no file named "HPW_PIPE" exists, or if the file exists but is empty. You must create and fill an MS-DOS file named "HPW_PIPE". This file is not created by HP BASIC.

Thus, a communications protocol is set up. MS-DOS programs can open and direct output to the "HPW_PIPE" file. The output can be read by the measurement coprocessor.

The following program segment shows one way of using this capability within a program.

```
DIM Result$[80]
  .
  .
OUTPUT 19; "DIR > HPW_PIPE"
ENTER 19; Result$
  .
  .
```

The string variable "Result$" will contain the first line of the output of the MS-DOS DIR command.

Background Mode and the Measurement Coprocessor

In background mode the measurement coprocessor operates on its own. An HP BASIC program runs on the coprocessor while the host PC is free to do other (MS-DOS) tasks. A major feature of background mode is that, since up to three coprocessors may be installed in the PC, more than one coprocessor can be operating at once. Only one coprocessor can be running HP BASIC in foreground (normal) mode, but all three can be running in background. Refer to chapter 11, "Multiple Coprocessors and Special Configurations," for further information.

Note A memory resident program, which occupies approximately 100 KB of PC memory (user RAM), is loaded for *each* measurement coprocessor that is running in background mode. Thus, each coprocessor that is running in background reduces by about 100 KB the amount of PC memory available for running MS-DOS applications in foreground. This memory requirement may place restrictions on the MS-DOS applications that you can run in foreground. For further information, refer to "Memory Requirements" in chapter 11.

In background mode the measurement coprocessor has unrestricted access to its own "built-in" HP-IB interface and to the DIO (GPIO and SRM) cards. The measurement coprocessor also has access to the speaker (for the BEEP command) and to *DFS* mass storage. (Some MS-DOS applications running in foreground may not be compatible with background access to mass storage.) However, if a program running in background attempts to access a PC HP-IB card, a serial port, or select code 15 mass storage; or attempts a graphics operation; BASIC will pause until foreground mode is restored.

Entering and Leaving Background Mode

As described previously, you can put the measurement coprocessor into background mode by either pressing "Backgnd" ((Ctrl) (F9)) or by executing:

```
OUTPUT 19;"BACKGROUND"
```

either within a program, or from the HP BASIC keyboard entry area of the screen.

The MS-DOS command line or PAM screen should appear (whichever was present when you started HP BASIC).

To re-enter foreground mode, type:

```
BASIC  (Enter)
```

from the MS-DOS command line. When the BASIC screen reappears, you'll find that the program is still running (unless it has completed its assigned tasks and has encountered the END statement).

Multiple Measurement Coprocessors

You can use the "card number" option to re-enter foreground mode for a specific measurement coprocessor. Execute one of the following:

```
BASIC 1 (Enter)
BASIC 2 (Enter)
BASIC 3 (Enter)
```

from the MS-DOS command line. However, don't use any of the /BOOT options. If you do, you will get an error message: "Unable to BOOT while in background mode."

Terminating a Background Process

If you leave the measurement coprocessor in background mode and you want to terminate its operation, execute the "BASIC" command to put it back in foreground mode, pause or stop the program, and then exit BASIC with (Ctrl) (F10). *This is the only orderly "shut down" procedure for a background process.*

Accessing the Display in Background Mode

Obviously, in background mode your program cannot actually display data on the CRT directly. However, your program can execute DISP and PRINT statements to the CRT without affecting program execution. You just won't be able to see the results until you restore foreground operation. For example, you could write a program that logs data from an HP-IB instrument and prints each reading to the screen while in background mode. When you return to foreground mode, you may examine the data on the screen, scrolling as needed.

If your program attempts a graphics command in background, however, the program will pause until you return to foreground mode. When you do, the graphics command will be executed.

Note

Upon entering background mode, all graphics display contents are lost. Only the alpha display is restored when you return to foreground mode.

Accessing Mass Storage in Background Mode

An HP BASIC program running in background can access *DFS* directories in the internal disk drives of your PC. Also, a program running in background can access an external HP-IB disk drive through select code 7. However, a program running in background *cannot* access LIF or HPW volumes in internal PC mass storage. *If your program attempts to access PC mass storage through select code 15 while running in background mode, BASIC will pause until foreground mode is restored.* For further information about mass storage and the DFS file system, refer to chapter 7, "File Systems and Mass Storage."

Caution

Be very careful about accessing flexible disks when in background mode. If an HP BASIC program running in background accesses a flexible disk, there is a danger that someone may remove the flexible disk from the drive (or place a different disk in the drive) prior to a disk access. If this happens, the results are unpredictable.

There are some restrictions on accessing DFS files from background mode. These result from the following considerations:

- An HP BASIC program and MS-DOS may both try to access the same file at the same time. (Refer to "Avoiding Collisions," below.)
- Various MS-DOS operations may change the current MS-DOS directory while an HP BASIC program is running in background. Thus, the HP BASIC program cannot predict or control what the current directory is. (Refer to "Avoiding Getting Lost.")
- Some MS-DOS applications running in foreground may not be compatible with background access to mass storage.

Avoiding Collisions

If an HP BASIC program and MS-DOS both attempt write accesses to the same file at the same time, the file could be corrupted. If HP BASIC or MS-DOS attempts to read a file while the other attempts to write to it, you may obtain unexpected results. (With multiple measurement coprocessors, the same problem could arise if two coprocessors attempt to access the same file simultaneously.) Thus, to protect the integrity of the file system, you need to avoid such collisions. The simplest way to do this is the least elegant. If an HP BASIC program accesses a file in background, just wait until it is done before accessing that file from MS-DOS. You could have the BASIC program "beep" when it is done, for example.

Unfortunately, this solution would deny some of the greatest advantages of background operation. The more sophisticated solution is to set up communications paths between MS-DOS and the measurement coprocessor, or between multiple coprocessors, so that they may avoid collisions by themselves. Several CSUBs (compiled subprograms) are provided with the measurement coprocessor software that allow you to set up such "smart" communications paths. Refer to *Integrating HP BASIC with MS-DOS Applications* for a complete discussion of this topic.

Note

When an HP BASIC program assigns an I/O path to a DFS file to be accessed while in background mode, you should specify *unbuffered* data transfer through the I/O path. To do this, the ASSIGN statement should be followed with a CONTROL statement that writes a "0" to Control Register 9 ("`CONTROL @File,9;0`", for example). Otherwise, if an MS-DOS application reads the file before the I/O path is closed, some of the data may still be buffered (not yet written to the file). Refer to "Additional I/O Path Registers for DFS" in chapter 7 for further information.

Avoiding Getting Lost

When you are writing an HP BASIC program that will run in background mode, you have to be careful about specifying where to access mass storage files. Otherwise, a file may be written to, or read from, an unexpected location. There are two safe ways to specify a file when in background mode:

1. *Use the current mass storage volume and directory for file access in background mode.* To access a file in the current mass storage, just specify a file name with no path or volume. The current mass storage is assumed.
2. *If you want to access a file in other than the current mass storage volume and directory, you must specify the full path name of the file.*

The current mass storage is specified with the MASS STORAGE IS statement, and can be a DFS volume and directory such as "\BLP\DATA:DOS,C". *HP BASIC maintains the current mass storage independent of any actions taken by MS-DOS.* Thus, you can safely access files in the current mass storage at any time, including in background mode.

Once the current mass storage has been specified, you can access files there with just the file name. For example, the statement:

```
CREATE BDAT "DATAFILE",13
```

creates a BDAT file in the current mass storage.

However, if you specify a volume, but not a directory in a file specifier, the *current MS-DOS directory on that volume* is assumed. For example, the statement:

```
CREATE BDAT "DATAFILE:DOS,D",13
```

will create a BDAT file in the current MS-DOS directory for hard disk volume "D:". The problem is that the current MS-DOS directory is maintained by MS-DOS, not by BASIC. Thus, even though the current MS-DOS directory for volume "D:" may be "D:\PROJECTS" when the BASIC program starts, MS-DOS may change the current directory while HP BASIC is running in background mode. The solution is simple. *In background mode, always use the full path name of a file, unless that file is in the current mass storage volume and directory.* For example, use the following statement:

```
CREATE BDAT "\PROJECTS\DATAFILE:DOS,D",13
```

instead of the previous one.

Examples

In the following program segment, line 350 puts HP BASIC into background mode, line 360 specifies the current mass storage, and line 370 creates the BDAT file "DATAFILE" there. Line 390 opens an I/O path to the file. The CONTROL statement on line 400 specifies *unbuffered* data transfer to the file. Line 410 outputs data to the file and line 420 closes the I/O path. Note that the current mass storage is used in all statements that access "DATAFILE".

```
340   ! Create file and store data from array "Trace_a"
350   OUTPUT 19; "BACKGROUND"
360   MASS STORAGE IS "\BLP\DATA:DOS,C"
370   CREATE BDAT "DATAFILE",13
380   !
390   ASSIGN @File TO "DATAFILE"
400   CONTROL @File,9;0
410   OUTPUT @File;Trace_a(*)
420   ASSIGN @File TO *
430   !
440   END
```

The following program segment does the same thing, but in this case the full path name of "DATAFILE" is used in each statement.

```
340   ! Create file and store data from array "Trace_a"
350   OUTPUT 19; "BACKGROUND"
360   !
370   CREATE BDAT "\BLP\DATA\DATAFILE:DOS,C",13
380   !
390   ASSIGN @File TO "\BLP\DATA\DATAFILE:DOS,C"
400   CONTROL @File,9;0
410   OUTPUT @File;Trace_a(*)
420   ASSIGN @File TO *
430   !
440   END
```

5

Using the Keyboard with HP BASIC

When HP BASIC is running, the measurement coprocessor takes control of the PC keyboard to emulate an HP BASIC workstation keyboard. This chapter describes how to use the measurement coprocessor with each supported keyboard. (The HP 82324A High-Performance Measurement Coprocessor and the HP 82300C Measurement Coprocessor provide identical keyboard functionality.)

Supported PC Keyboards

The measurement coprocessor supports several PC keyboards. These keyboards are fully supported as *system* keyboards. That is, the keyboards offer full control of the computer both from MS-DOS and from HP BASIC. For each of the PC keyboards, while operating in HP BASIC the measurement coprocessor software emulates the functions of the HP 9000 Series 300 style ITF keyboard, but maps these functions to the keys available on the particular PC keyboard. (For more information about the ITF keyboard, refer to "Other Supported Keyboards.")

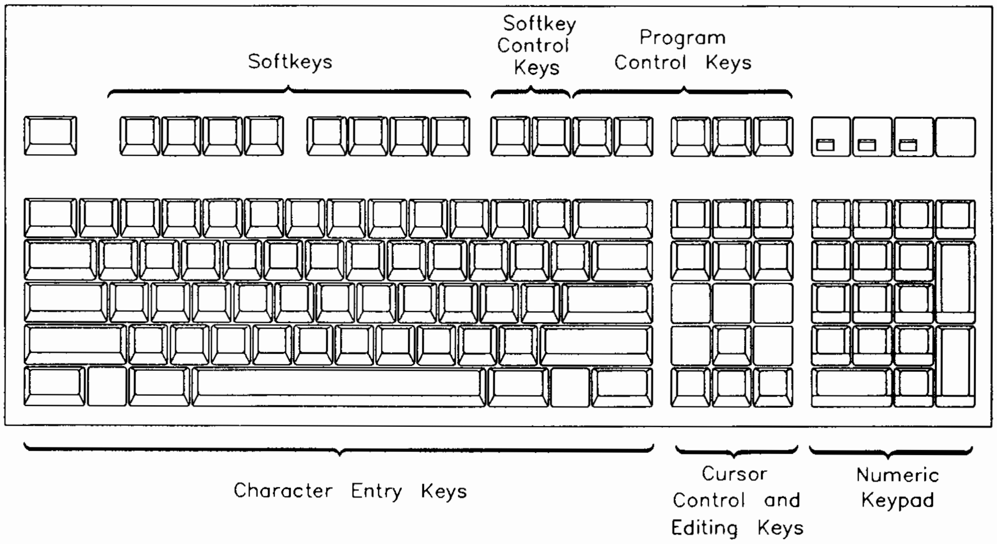
The supported PC keyboards are as follows

- *Vectra PC Industry-Standard Keyboard.* The industry-standard keyboard is also known as the "enhanced 101/102-key keyboard." (The US-English version has 101 keys, while the UK-English and other versions have 102 keys.) This keyboard is used by all current Vectra PCs including the Vectra 286/12, QS, and RS PCs. *Most AT-compatible computers now available use the industry-standard keyboard.* This keyboard is described in the next section.

- *Old-Style Vectra PC Keyboard.* The original Vectra PC (pre-1988) used an HP-HIL keyboard which is described in "Using the Old-Style Vectra PC Keyboard" later in this chapter.
- *Older AT-Style Keyboard.* Many AT-compatible computers of pre-1988 vintage used an 84-key keyboard, which is described in "Using the Older AT-Style Keyboard" later in this chapter.

Using the Vectra PC Industry-Standard Keyboard

All current Vectra PC models use the industry-standard "enhanced 101/102-key" keyboard. This keyboard is also used by most currently available AT-compatible PCs. The keys on this keyboard are arranged into the following functional groups:



This section provides you with key definitions for the Vectra PC industry-standard keyboard.

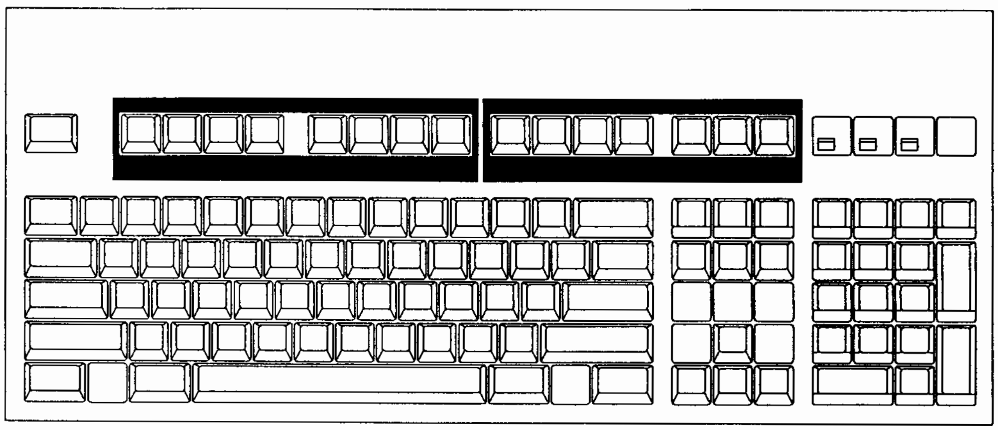
Note

If you want to see the action of the keys demonstrated in the following steps, boot your HP BASIC system in the normal manner. Then type `SCRATCH` (Enter) before proceeding.

The "Num Lock," "Caps Lock," and "Scroll Lock" lights on the keyboard are *not* supported by HP BASIC. The states of these lights are not indicative of the keyboard state for HP BASIC. However, HP BASIC displays a "Caps" indicator on the screen.

HP BASIC Keyboard Overlays

Two keyboard overlays designed for the industry-standard keyboard are included with your measurement coprocessor. Place the overlays on the keyboard as shown below (HP part number 5180-1552 on the left, 5180-1558 on the right):



Character Entry Keys

The character keys are arranged like a typewriter, but have some added features.

Caps lock — This key sets the unshifted keyboard to either uppercase (the default after HP BASIC is booted) or lowercase. HP BASIC displays the keyboard mode when you press the Caps lock key.

Type in a few words, then press the Caps lock key and continue typing. Notice the case change. Press Ctrl End when you are finished.

Shift — You can use the Shift key to alternate between entering standard uppercase and lowercase letters. This is exactly the same as a typewriter.

Enter — The Enter key has three functions:

- When a program that is executing prompts you for data, respond by typing the requested data and then pressing Enter. This signals the program that you have provided the data it requested and it can continue.
- When typing in lines of a program the Enter key is used to store each line of program code.
- After typing in a command the Enter key causes the command to be executed.

Type `EDIT` and press Enter. Notice the number `10` is displayed on the screen. This is the line number of the first line of an HP BASIC program. The computer is waiting for you to type in the line. Type:

```
!FIRST LINE
```

and press Enter. The computer accepts the statement as a program line and displays `20` in preparation for the next line. Press F1 when you are finished.

Alt — When pressed along with another key, this key allows you to generate the rest of the full 256-bit character set from the main typewriter section on Standard and European keyboards.

(Tab)	This key moves the cursor forward to preset tabs. Pressing (Shift)(Tab) moves the cursor backward to preset tabs. Before (Tab) can be used, a tab must be set. Tabs are set and cleared with system menu softkeys. This will be explained in the section entitled "System Softkeys."
(Ctrl)	The control key works like the (Shift) key to access a set of standard control characters, such as line-feed and form-feed. These characters are useful to you for controlling some devices and communicating with other computers.
(Shift)(Enter)	The Select function ((Shift)(Enter)) beeps but performs no function unless it is program-defined.

Cursor Control Keys

The cursor-control keys move the display cursor.

(▲)(▼) (Shift)(▲) (Shift)(▼)	The (▲) and (▼) keys allow you to scroll lines in the output area up and down. Shifted, the keys allow you to jump to the top and bottom of the output area.
(◀)(▶) (Shift)(◀) (Shift)(▶)	The (◀) and (▶) keys allow you to move horizontally along a line. Shifted, they allow you to jump to the left and right limits of a line.
(←)	The backspace key works just like the (◀) key.

(Home)
(Shift)(Home)

The (Home) key positions the print position at the beginning position on the page. The shifted (Home) key places the print position at the beginning of the first empty line in the display (scrolls up if necessary). In edit mode, pressing this key (shifted or unshifted) causes the computer to beep. To verify operation of the (Home) key, press (Ctrl)(End). Then type **PRINT "SOMETHING"** and press (Enter); repeat twice. You should now have the following display:

```
SOMETHING
SOMETHING
SOMETHING
```

Press the (Home) key (unshifted).

Type **PRINT "ANY "** and press (Enter). Your display should look like this:

```
ANY THING
SOMETHING
SOMETHING
```

Press (Ctrl)(End).

(Page Up)
(Page Down)

In normal mode, pressing the (Page Up) key causes the display to scroll down one page and pressing the (Page Down) key causes the display to scroll up one page. In edit mode, these keys move the display one-half page.

To test the horizontal movement of the cursor, type a few words and press the shifted and unshifted (◀) and (▶) keys. Notice that the cursor cannot be moved beyond the characters you have typed. Press (Shift)(End) when finished.

To test the vertical movement of the cursor, type **EDIT** and press (Enter). Now type the following lines, pressing (Enter) after each line (the first line may be there already, so just press (Enter) to accept it):

```
10  !FIRST LINE
20  !SECOND LINE
30  !THIRD LINE
40  !FOURTH LINE
```

Try the shifted and unshifted (▲), (▼), and (Home) keys. Then try the (Page Up) and (Page Down) keys. When you're done, press (Pause) to exit. Then type **SCRATCH** and press (Enter) to clear memory.

Numeric Keypad

The numeric keypad provides a convenient way to enter numbers and perform arithmetic operations. Simply type in the arithmetic expression you want to evaluate and press (Enter). The result is displayed in the lower-left corner of the screen.

Type in the following problem using the numeric keypad:

```
(26+14)/4
```

Now press (Enter) to perform the calculation. The answer, 10, is displayed in the lower-left corner of the screen.

Editing Keys

The editing keys put easy character editing and line editing at your fingertips.

(Shift)(Insert)

Pressing (Shift) and (Insert) inserts a new line above the cursor's current position (edit mode only).

Type EDIT, then press (Enter). Type in this line (if it isn't already there):

```
10 !FIRST LINE
```

Now, with the cursor somewhere on line 10, press (Shift)(Insert). Notice that a new line number (1) is inserted before line 10. Press (Pause) when finished.

(Shift)(Delete)
Delete line

Pressing (Shift)(Delete) deletes the line containing the cursor (edit mode only).

Type EDIT, then press (Enter). Position the cursor to the line:

```
10 !FIRST LINE
```

and press (Shift)(Delete). The line is removed. To restore the line, press (Shift)(Page Up), then press (Enter) to enter it into the program. Press (Pause) to exit edit mode.

Insert
Insert char

Pressing Insert sets insert mode, allowing you to insert characters to the left of the cursor. Press the key a second time to cancel insert mode.

Carefully type the following line exactly as shown:

```
THIS IS A TEST .
```

Position the cursor under the period and press Insert. Now type:

```
OF INSERT MODE
```

and press Insert again. The line should now look like this:

```
THIS IS A TEST OF INSERT MODE.
```

The new characters were inserted to the left of the period. Press Shift End when finished.

Delete
Delete char

Pressing Delete deletes the character at the cursor's position.

Type a few words and experiment with Delete, positioning the cursor at various places on the line. Notice that if you hold the key down, characters are deleted until you release it. Delete all of the characters you typed.

End
Clear to End

Pressing End clears from the current cursor position to the end of the line.

Shift End
Clear Line

Pressing Shift End clears the keyboard line and the message/results line. (In EDIT mode, it clears the line the cursor is on.)

Type in a few words and use the ◀ key to position the cursor in the middle of the line. Press End to clear to the end of the line. Press Shift End to clear the rest of the line.

Ctrl End
Clear screen

Pressing Ctrl End clears the entire alpha screen.

Type the following HP BASIC command:

```
PRINT "PUT THIS MESSAGE IN THE
OUTPUT AREA."
```

Now press Enter to execute it. Press Shift Page Up to recall the command, and press Enter again. Repeat this step several times to fill the screen with messages. Now press Ctrl End to erase all lines at once.

Program Control Keys

The following keys allow you to control execution of the program stored in the computer's memory.

Scroll Lock Clr I/O	Pressing Scroll Lock pauses program execution when the computer is performing or trying to perform an I/O operation. Press Scroll Lock instead of Shift Pause (Stop) when the computer is hung up on an I/O operation since Shift Pause works only after the computer finishes the current program line. Pressing Scroll Lock cancels the I/O operation and pauses the program at the current line.
Shift Scroll Lock Reset	Pressing Shift Scroll Lock (Reset) pauses program execution immediately without erasing the program from memory. The HP BASIC Reset message indicates the computer is ready for your command.
Shift Page Up Recall	Pressing Shift Page Up recalls the last line that you entered, executed, or deleted. Several previous lines can be recalled this way. Recall is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again. Type: `PRINT "1"` Enter to print the number 1 on the screen. Now press Shift Page Up to recall the print statement. Edit the statement to print the number 2 by positioning the cursor under the 1 and typing 2 over it. Press Enter again. Now press Shift Page Up several times to see all of the statements it remembers. Then press Ctrl End to clear the screen when you are finished. Pressing F8 in the System menu performs the same recall function as Shift Page Up.
Shift Page Down	Press Shift Page Down to move forward through the recall stack.
Pause	Pressing Pause pauses program execution after the current line. Pressing `Continue` (F2) in the System menu resumes program execution from the point where it was paused.

Shift Pause Stop	Pressing Shift Pause stops program execution after the current line. To restart the program, press RUN (F3) in the System menu.
Shift Print Screen Step	Shift Print Screen allows you to execute one program line at a time. This is particularly useful for debugging programs.
Print Screen Print All	Print All (Print Screen) turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device.

Press Print All Print Screen once to set printall on and again to set printall off. The display's output area is the default printall device at power-up.

Press Print Screen (Print All) to turn on printall mode. Now type in the following command:

```
PRINT "THIS IS A KEYBOARD
OPERATION"  Enter
```

Both the PRINT command and the message itself are displayed on the screen, which is the default printall device. Now type:

```
THIS WILL CAUSE AN ERROR  Enter
```

Because this is not an executable HP BASIC statement, an error message is displayed at the bottom of the screen and in the printall area at the top of the screen. A log of all commands typed and executed at the keyboard, along with any error messages, is thus produced. Press Ctrl End to clear the display, and press Print Screen (Print All) to turn off printall mode.

Ctrl F9 Background	Pressing Ctrl F9 places HP BASIC in background operation. This is identical to executing OUTPUT 19;"BACKGROUND" from HP BASIC. Refer to chapter 4 for more information on background operation.
Ctrl F10 EXIT	Pressing Ctrl F10 terminates HP BASIC and returns your computer to MS-DOS.

F11 Alpha	Pressing F11 once turns on the alphanumeric display. Pressing it a second time turns off the graphics display. This key function requires that the GRAPH binary be loaded.
Shift F11 Dump Alpha	Pressing Shift F11 prints a complete copy of the alpha display on the default printer.
F12 Graphics	Pressing F12 once turns on the graphics display. Pressing it the second time turns off the alphanumeric display.
Shift F12 Dump Graph	Pressing Shift F12 prints a complete copy of the graphics display on the default printer.
	Both Graphics and Dump Graph key functions require that the GRAPH binary extension file be loaded.

Softkeys and Softkey Control

There are eight softkeys (labeled F1 through F8) and two keys (F9 and F10) that control the definitions of the softkeys (MENU and SYSTEM).

When the HP BASIC system is booted, the softkeys default to System mode. The System mode menu appears at the bottom of your display. System softkeys are defined following control key definitions. In addition to the System mode, there are also three User modes: User 1, User 2, and User 3. You can change the functions and labels of the User menu softkeys. Refer to "Editing Softkeys" later in this chapter.

Softkey Control Keys

There are three control keys for the System, User, and Menu functions.

F9 Menu	Pressing F9 toggles the softkey labels (turns them on if they're off and turns them off if they're on).
Shift F9	Pressing Shift F9 increments user mode and menu *if* user mode is on.
F10 System	Pressing F10 causes softkeys to assume System mode. The System menu is displayed if the Menu key (F9) is toggled to the on position.
Shift F10 User	Pressing Shift F10 puts the softkeys in User 1 mode. The User 1 menu is displayed if the Menu key (F9) is toggled to the on position.

User menus are blank unless the KBD language extension binary is loaded.

Now let's look at the control keys.

First we want to get the System mode selected and menu displayed. If the System menu is displayed, continue with the next paragraph. If it is not displayed, press F10 (System). If it is still not displayed, press F9 (Menu).

With the System menu displayed, press F9 (Menu) several times. The system menu display should go on and off. Leave the System menu displayed and press Shift F10 (User). The User 1 menu should appear on your display.

Press Shift F9 (Shift Menu) several times. The displayed menus should rotate successively through the three User menus (User 1 → User 2 → User 3 → User 1 → User 2 →, etc.).

Press F9 (menu) several times and the last User menu goes on and off. Leave the User menu on.

Finish this exercise by pressing F10 (System) to get your softkeys back in System mode.

System Softkeys

The following paragraphs define the eight System softkeys.

(F1) Step	`Step` allows you to execute one program line at a time. This is particularly useful for debugging programs.
(F2) Continue	`Continue` resumes program execution from the point where it was paused by (Pause).
(F3) Run	`Run` starts a program running from the beginning.
(F4) Print All	The `Print All` key ((F4)) turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device. Press (F4) once to turn printall on. Press it again to turn printall off. A message appears on the screen indicating whether printall is on or off.
(F5) and (Shift)(F5) Set Tab and Clr Tab	`Set Tab` ((F5)) sets a tab at the cursor's current position. Tabs remain in effect until cleared by either `Clr Tab` ((Shift)(F5)) or the SCRATCH A statement `Clr Tab` ((Shift)(F5)) clears a tab previously set at the cursor's position. Press the space bar to move the cursor forward a few spaces and press (F5) (`Set Tab`). Move the cursor back several spaces using (◀), then pres (Tab). Move the cursor forward several more spaces with the space bar, then press (Shift)(Tab). To clear the tab, move the cursor to the unwanted tab position and press (Shift)(F5) (`Clr Tab`). Press (Shift)(End) when finished.

(F6)
Display Functions

Pressing Display Fctns ((F6)) turns on the display functions mode, allowing you to see special control characters (form-feed and carriage control, for example) on the screen. Pressing (F6) again turns the display functions mode off. An asterisk (*) appears, indicating that display functions mode is on.

Type the following line:

```
PRINT "DISPLAY FUNCTIONS
MODE OFF"
```

and press (Enter). Notice the display at the top of the screen. Now press Recall ((F8)) to recall the line, and edit it to read:

```
PRINT "DISPLAY FUNCTIONS ON"
```

Press Display Fctns ((F6)) and then press (Enter). Notice that the carriage return and line feed control characters are now displayed. Press Display Fctns ((F6)) again to turn off the display functions mode. Press Clear Screen ((Ctrl)(End)) when you are finished.

(F7)
Any Char

Any Char ((F7)) is used to find any ASCII character. First press (F7) (Any Char). The following message appears above the menu:

```
Enter 3 digits, 000 to 255
```

Enter a three-digit number from 000 through 255 representing the decimal equivalent of an ASCII character. The computer automatically displays the character on the screen. For a list of characters and their equivalent decimal values, see the US ASCII Character Codes table in the "Useful Tables" appendix of the BASIC *Language Reference* manual.

Press (F7) (Any Char), then type 065, which is the decimal equivalent of "A". The display line now displays "A". Press (Shift)(End) to erase it.

F8 or
Shift Page Up
Recall

The **Recall** softkey (F8 or Shift Page Up) acts just like system control key F10 described earlier in this section. **Recall** recalls the last line that you entered, executed, or deleted. Several previous lines can be recalled this way. **Recall** is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again.

Type:

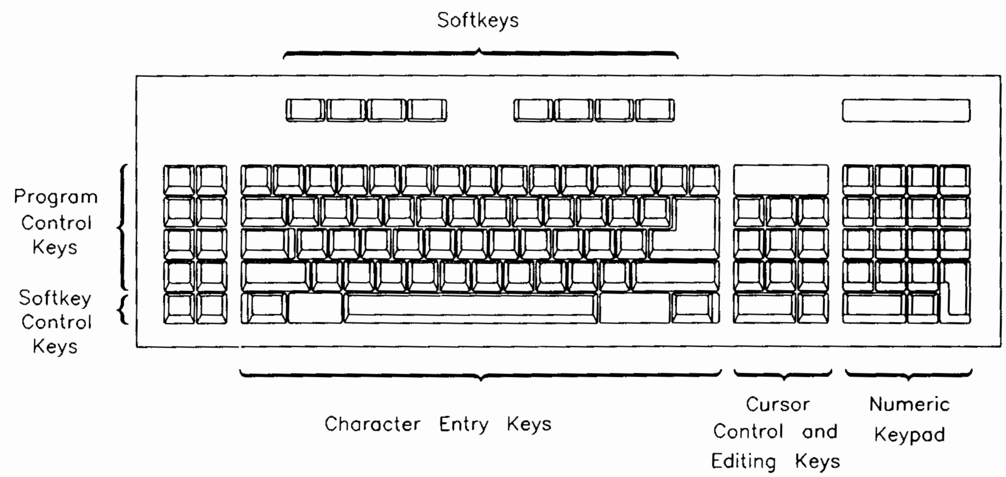
```
PRINT "1"
```

and press Enter to print the number 1 on the screen. Now press **Recall** (F8) to recall the PRINT statement. Edit the statement to print the number 2 by positioning the cursor under the number 1 and entering the number 2. over it. Press Enter again. Now press **Recall** (F8) to see all of the statements it remembers.

Note that **Recall** goes backward through the queue. Pressing Shift F8 (or Shift Page Down) allows you to cycle forward through the queue until the last line entered, executed, or deleted is displayed.

Using the Old-Style Vectra PC Keyboard

Vectra PC computers prior to the Vectra CS, ES, QS, and RS (pre-1988) used the HP-HIL keyboard shown below. The keys on the old-style Vectra PC keyboard are arranged into the following functional groups:



This section provides you with key definitions for the old-style Vectra PC keyboard.

Note

If you want to see the action of the keys demonstrated in the following steps, boot your HP BASIC system in the normal manner. Then type SCRATCH Enter before proceeding.

The "Caps Lock," "Num Lock," and "Scroll Lock" lights on the keyboard are *not* supported by HP BASIC. These lights are inactive when HP BASIC is running. However, HP BASIC displays a "Caps" indicator on the screen.

HP BASIC Keyboard Overlays

Two keyboard overlays designed for the old-style Vectra PC keyboard are included with your measurement coprocessor. Place the overlay with HP part number 5180-1550 over the eight softkeys at the the top of the keyboard. Place the overlay with HP part number 5180-1551 over the ten softkeys on the left side of the keyboard.

Note Make sure that you use overlay number 5180-1551, *not* number 5180-1562 for the left-side softkey pad. The 5180-1562 overlay is intended for the older AT-style (84-key) keyboard.

Let's look at the individual functional groups of the keyboard.

Character Entry Keys

The character entry keys function as described in "Using the Vectra PC Industry-Standard Keyboard." (The key locations may differ.)

Cursor Control Keys

The cursor control keys function as described in "Using the Vectra PC Industry-Standard Keyboard," except that you press (F3) to exit edit mode, rather than (Pause). (The key locations may differ.)

Numeric Keypad

The numeric keypad functions as described in "Using the Vectra PC Industry-Standard Keyboard." (The key locations may differ.)

Editing Keys

The editing keys function as described in "Using the Vectra PC Industry-Standard Keyboard," except that you press (F3) to exit edit mode, rather than (Pause). (The key locations may differ.)

Program Control Keys

The following keys allow you to control execution of the program stored in the computer's memory.

F1
Clr I/O — Pressing F1 pauses program execution when the computer is performing or trying to perform an I/O operation. Press F1 instead of Shift F3 (Stop) when the computer is hung up on an I/O operation since Shift F3 works only after the computer finishes the current program line. Pressing F1 cancels the I/O operation and pauses the program at the current line.

Shift F1
Reset — Pressing Shift F1 (Reset) pauses program execution immediately without erasing the program from memory. The HP BASIC Reset message indicates the computer is ready for your command.

F2 or
Shift Pg Up
Recall — Press either F2 or Shift Pg Up to recall the last line that you entered, executed, or deleted. Several previous lines can be recalled this way. Recall is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again. Type:

```
PRINT "1"   Enter
```

to print the number 1 on the screen. Now press F2 to recall the print statement. Edit the statement to print the number 2 by positioning the cursor under the 1 and typing 2 over it. Press Enter again. Now press F2 several times to see all of the statements it remembers. Then press CTRL End to clear the screen when you are finished.

Pressing f8 in the System menu performs the same recall function as F2.

Shift F2 or Shift Pg Dn	Press either Shift F2 or Shift Pg Dn to move forward through the recall stack.
F3 Pause	Pressing F3 pauses program execution after the current line. Pressing `Continue` (f2) in the System menu resumes program execution from the point where it was paused.
Shift F3 Stop	Pressing Shift F3 stops program execution after the current line. To restart the program, press `RUN` (f3) in the System menu.
F5 Step	F5 allows you to execute one program line at a time. This is particularly useful for debugging programs.
Shift F5 Print All	Print All (Shift F5) turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device.

Press Shift F5 to turn on printall mode. Now type in the following command:

```
PRINT "THIS IS A KEYBOARD
OPERATION"  Enter
```

Both the PRINT command and the message itself are displayed on the screen, which is the default printall device. Now type:

```
THIS WILL CAUSE AN ERROR  Enter
```

Because this is not an executable HP BASIC statement, an error message is displayed at the bottom of the screen and in the printall area at the top of the screen. A log of all commands typed and executed at the keyboard, along with any error messages, is thus produced. Press CTRL End to clear the display, and press Shift F5 to turn off printall mode.

F7 Alpha	Pressing F7 once turns on the alphanumeric display. Pressing it a second time turns off the graphics display. This key function requires that the GRAPH binary be loaded.

(Shift)(F7) Dump Alpha	Pressing (Shift)(F7) prints a complete copy of the alpha display on the default printer.
(F8) Graphics	Pressing (F8) once turns on the graphics display. Pressing it the second time turns off the alphanumeric display.
(Shift)(F8) Dump Graph	Pressing (Shift)(F8) prints a complete copy of the graphics display on the default printer.
	Both Graphics and Dump Graph key functions require that the GRAPH binary extension file be loaded.
(CTRL)(F9) Background	Pressing (CTRL)(F9) places HP BASIC in background operation. This is identical to executing OUTPUT 19;"BACKGROUND" from HP BASIC. Refer to chapter 4 for more information on background operation.
(CTRL)(F10) EXIT	Pressing (CTRL)(F10) terminates HP BASIC and returns your computer to MS-DOS.

Softkeys and Softkey Control

The HP BASIC softkeys and softkey control keys function as described in "Using the Vectra PC Industry-Standard Keyboard," except for the following:

- The keys labelled (f1) through (f8) across the *top* of the keyboard are used as HP BASIC softkeys. (The keys labelled (F1) through (F8) on the left side of the keyboard are used as program control keys.)
- The softkey control keys ((F9) and (F10)) are located on the lower left side of the keyboard.

Using the Older AT-Style Keyboard

Prior to 1988 many AT-compatible computers (other than the HP Vectra PC) used an 84-key keyboard. This keyboard is very similar to the old-style Vectra PC keyboard, but lacks the eight softkeys across the top. The ten keys (F1 through F10) along the left side of the keyboard are the same as for the old-style Vectra PC keyboard.

If your computer uses the older AT-style (84-key) keyboard, you will need to perform some additional installation steps *after* you complete the software installation procedures in chapter 2 of this manual:

1. Return your PC to MS-DOS (indicated by the > prompt) if it is not already there. If you are in HP BASIC, you can return to MS-DOS by pressing Ctrl F10.
2. Insert your "Manual Examples, LIFINIT, and Selected CSUBs" disk in drive A:.
3. Copy the file OLDIBM.CON to the file BLP.CON in your HP BASIC directory. For example, if you have installed HP BASIC in a directory called C:\BLP on your hard disk drive, you would enter the following command:

   ```
   COPY A:\OLDIBM.CON C:\BLP\BLP.CON   Enter
   ```

4. Install the keyboard overlay with HP part number 5180-1562, included with your measurement coprocessor, over the ten soft keys on the left side of your keyboard. (Don't use overlay number 5180-1551 — it is intended for the old-style Vectra PC keyboard.)
5. Refer to "Using the Old-Style Vectra PC Keyboard" for information about the individual key functions.

Other Supported Keyboards

Your PC keyboard functions as the *system keyboard*. That is, it can control your computer in the MS-DOS environment as well as from HP BASIC. However, some users prefer to use a workstation-style keyboard for HP BASIC. Two such keyboards are available:

- The HP 46021A ITF Keyboard (an HP 9000 Series 300 ITF keyboard).
- The HP 98203C Keyboard (an HP 9000 Series 200 style keyboard).

Either of these keyboards can be used as a *secondary keyboard* for HP BASIC operation only. If your Vectra PC has a built-in HP-HIL port, you can connect either of these keyboards to it. If you have a Vectra PC that has no HP-HIL port, you will need an HP-HIL card. (These keyboards are not supported for non-HP computers.)

Note

You can use the added HP-HIL keyboard only from HP BASIC, not from the MS-DOS environment. Thus, you'll normally need to leave the PC keyboard connected in order to start up the system.

You can set up your system to automatically start HP BASIC at power up. (Refer to "Autostarting HP BASIC" in chapter 3 for further information.) However, if you exit BASIC you will still need a system keyboard to re-enter BASIC.

The keyboard functions of the HP 46021A ITF Keyboard and the HP 98203C Keyboard are described in *Using the BASIC 5.0/5.1 System.*

Editing Softkeys

As mentioned previously in this chapter, the eight BASIC softkeys (F1 through f8) can be used in System mode and in three User modes, User 1, User 2, and User 3. HP BASIC provides pre-defined softkey labels for all of these modes. The softkey definitions are fixed in System mode. However, you can change the definitions of the softkeys in the User modes. This section tells how to edit the User mode softkeys.

Note In order to redefine softkeys, the KBD binary is required. The measurement coprocessor INSTALL program automatically loads all binaries, so KBD is normally present.

Changing Softkey Definitions

Example 1

First, as an example of using a softkey as a typing aid, we will redefine softkey F1 to produce the characters "My very own keystrokes" on the display whenever you press the key. Follow these steps:

Step 1:. Get into the User 1 softkey menu and press the EDIT key (F1), and then press F1 again followed by Enter.

The system now displays the following:

`k #EDIT`	*Displayed on the keyboard input line.*
`Editing key 1`	*Displayed on the system message line.*

The keyboard input line indicates the current definition of the softkey (EDIT) and the system message line indicates that you can now edit the definition of [F1].

Step 2:. Press [Shift] [End] (Clear Line) to clear the key's current definition.

Step 3:. Enter the new definition (type in the characters):

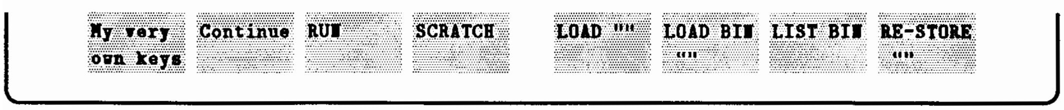
```
My very own keystrokes
```

Step 4:. Exit the softkey edit mode.

After you have finished modifying the softkey's definition, press [Enter] to exit the softkey edit mode and update the softkey's definition. (If you want to retain the current definition of the softkey, press [Pause] instead.)

This completes the softkey re-definition. Whenever you press [F1], the characters "`My very own keystrokes`" will be displayed. (Press [Shift] [End] to clear the line again.)

The softkey labels now appear as follows:



Note that the last six characters are not displayed in the key label because there isn't room.

Example 2

Softkey typing aids are very useful, but in many instances you will want the softkey to execute a command directly. For example, let's redefine (F2) to execute the LIST command.

Step 1:. Type:

```
EDIT KEY 2  [Enter]].
```

(You will have to type "EDIT KEY 2" this time because we redefined (F1) in the last example.)

Step 2:. Press (Shift) (End) (Clear Line) to clear the key's current definition.

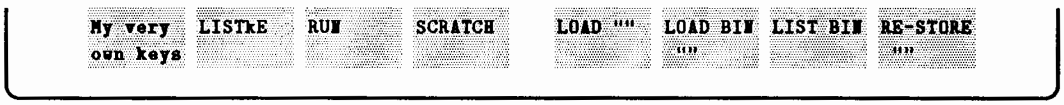
Step 3:. Enter the new definition as follows:

Type:

LIST and press (Ctrl) (Enter) followed by (Enter) again.

The notation (Ctrl) (Enter) indicates that you are to hold down the (Ctrl) key and then press the (Enter) key, then release the (Enter) key, and finally release the (Ctrl) key. The (ctrl) key tells the system that you don't want to execute the following key's function, but you want to enter a sequence of characters on the screen. The sequence begins with the inverse-video "k" character, CHR$(255), followed by another character. For example, (Ctrl) (Enter) produces "kE". *Note that on some monochrome displays inverse-video characters are not visible.*

The softkey labels now appear as follows:



When you press (F2), the LIST command is executed.

To restore the original (power-up) key definitions, just execute the statement:

```
LOAD KEY
```

Saving Softkey Definitions

Thus far we have discussed changing softkey definitions. However, if you don't save these definitions to a file, the original, system-defined softkey definitions will reappear after the measurement coprocessor is re-booted.

If you want to save the current softkey definitions in a file called "MyKeys", you could use the statement:

```
STORE KEY "MyKeys"
```

If the file already exists, you must use the command:

```
RE-STORE KEY "MyKeys"
```

To retrieve the key definitions from the file, use:

```
LOAD KEY "MyKeys"
```

Keyboard Mapping

The following table shows the relationships between the Series 200/300 keyboards and the old-style Vectra PC and Vectra PC industry-standard keyboards.

Keyboard Mapping

Function	HP 98203C Keyboard (Series 200)	HP 46021A ITF Keyboard (Series 300)	Old-Style Vectra PC Keyboard	Vectra PC Industry-standard Keyboard
ALPHA	Alpha	Unlabeled Key 2	F7	F11
ANY CHARACTER	Any Char	(System) f7	(System) f7	(System) F7
BACKGROUND	*	*	CTRL F9	Ctrl F9
CLEAR TO END	Clr▸End	Clr Line	End	End
CLEAR I/O	Clr I/O	Break	F1	Scroll Lock
CLEAR LINE	Clr Ln	Shift Clr Line	Shift End	Shift End
CLEAR SCREEN	Clr Scr	Clr Disp	CTRL End	Ctrl End
CLEAR TAB	Clr Tab	(System) Shift f5	(System) Shift f5	(System) Shift F5
CONTINUE	Continue	(System) f2	(System) f2	(System) F2
DELETE CHARACTER	Del Chr	Del Chr	Del	Delete
DELETE LINE	Del Ln	Del Ln	Shift DEL	Shift Delete
DISPLAY FUNCTIONS	Display Fctns	(System) f6	(System) f6	(System) F6

Keyboard Mapping (continued)

Function	HP 98203C Keyboard (Series 200)	HP 46021A ITF Keyboard (Series 300)	Old-Style Vectra PC Keyboard	Vectra PC Industry-standard Keyboard
DUMP ALPHA	Dump Alpha	Shift unlabeled key 2	Shift F7 or Shift * (numpad)	Shift F11
DUMP GRAPHICS	Dump Graphics	Shift unlabeled key 3	Shift F8	Shift F12
EDIT	Edit	(user 1) f1	(user 1) f1	(user 1) F1
EXECUTE	EXECUTE	†	†	†
EXIT HP BASIC	*	*	CTRL F10	Ctrl F10
SOFTKEY 0	k0	†	†	†
SOFTKEY 1	k1	f1	f1	F1
SOFTKEY 2	k2	f2	f2	F2
SOFTKEY 3	k3	f3	f3	F3
SOFTKEY 4	k4	f4	f4	F4
SOFTKEY 5	k5	f5	f5	F5
SOFTKEY 6	k6	f6	f6	F6
SOFTKEY 7	k7	f7	f7	F7
SOFTKEY 8	k8	f8	f8	F8
SOFTKEY 9	k9	†	†	†
GRAPHICS	Graphics	Unlabeled Key 3	F8	F12

Keyboard Mapping (continued)

Function	HP 98203C Keyboard (Series 200)	HP 46021A ITF Keyboard (Series 300)	Old-Style Vectra PC Keyboard	Vectra PC Industry-standard Keyboard
INSERT CHARACTER	Ins Chr	Ins Chr	Ins	Insert
INSERT LINE	Ins Ln	Ins Ln	Shift Ins	Shift Insert
MENU	†	Menu	F9	F9
PAUSE	Pause	Stop	F3	Pause
PRINT ALL	Prt All	(System) f4	(System) f4 or Shift F5	(System) F4 or Print Screen
RECALL	Recall	Unlabeled Key 1	F2 or (System) f8 or Shift Pg Up	(System) F8 or Shift Page Up
RECALL FORWARD	Shift Recall	Shift Unlabeled Key 1	Shift F2 or Shift Pg Dn or (System) Shift f8	Shift Page Down or (System) Shift F8
RESET	Reset	Shift Break	Shift F1	Shift Scroll Lock
RESULT	Result	Unlabeled Key 4	F4	RES (keyword)
RUN	Run	(System) f3	(System) f3	(System) F3
SET TAB	Set Tab	(System) f5	(System) f5	(System) F5
SELECT	†	Select	Shift Enter	Shift Enter

Keyboard Mapping (continued)

Function	HP 98203C Keyboard (Series 200)	HP 46021A ITF Keyboard (Series 300)	Old-Style Vectra PC Keyboard	Vectra PC Industry-standard Keyboard
STEP	Step	(System) f1	F5 or (System) f1	(System) F1 or Shift Print Screen
STOP	Stop	Shift Stop	Shift F3	Shift Pause
SYSTEM KEYS	†	System	F10	F10
USER KEYS	†	User	Shift F10	Shift F10

* No equivalent in Series 200/300.

† Cannot generate keystroke from this keyboard.

Using Localized Keyboards

The default keyboard for the measurement coprocessor is the US ASCII keyboard. However, several localized Vectra PC keyboards are supported.

Note

If you are using a keyboard other than US ASCII, it may use some of the extended ASCII characters (ASCII characters 128 – 255) in its localized alphabet (as shown on the key caps). You can type these characters directly from the keyboard, but only if these characters are enabled:

To enable these characters, execute: `CONTROL CRT,11;1`

To return to default display mapping, execute: `CONTROL CRT,11;0`

The following table lists the supported localized keyboards:

Vectra PC Industry-Standard Keyboard	**Old-style Vectra PC Keyboard**
US ASCII	US ASCII
French	French
German	German
Italian	Italian
Spanish	Spanish
Swedish	Swedish
UK English	UK English
Swiss	
Finnish	

The rest of this section covers three methods for selecting a keyboard localization.

Changing the MS-DOS Country Code

You can change the country code by including a COUNTRY command in the CONFIG.SYS file. This changes the keyboard localization for MS-DOS as well as for HP BASIC. The COUNTRY command syntax depends on the version of MS-DOS that you are using. Refer to the "System Configuration" chapter in your HP Vectra *MS-DOS User's Reference* manual.

Setting the BLPLANG Environment Variable

You can also select the keyboard localization for HP BASIC by setting the localization environment variable, BLPLANG, before you start BASIC. The BLPLANG environment variable setting takes effect when you next boot the measurement coprocessor. *The BLPLANG setting overrides the MS-DOS country code, but only for HP BASIC.*

To set the environment variable, execute the following statement from the MS-DOS command line:

`SET BLPLANG=`*XX*

Where *XX* is the localization mnemonic from the table below:

Language	Mnemonic
US ASCII	US
French	FR
German	GR
Italian	IT
Spanish	SP
Swedish	SW
UK English	UK
Swiss (Ger.)	SG
Finnish	FI

For example, to set the localization environment variable to "French," starting from the MS-DOS prompt, type:

`SET BLPLANG=FR` (Enter)

Now start BASIC with the statement:

`BASIC /BOOT` (Enter)

Note that the environment variable only takes effect when the measurement coprocessor is booted. Thus, you could start BASIC with the statement:

`BASIC` (Enter)

but only if the measurement coprocessor has not already been booted. The `/BOOT` option forces the measurement coprocessor to be booted in any event, and causes the new environment variable to take effect.

If you want to execute the SET BLPLANG statement automatically each time you turn on or boot your computer, include the statement in your AUTOEXEC.BAT file.

Specifying Localization When Starting BASIC

You can also specify a localization option when you start HP BASIC. To do this, start BASIC from the MS-DOS prompt (for directory C:\BLP) by typing:

`BASIC /`*XX* (Enter)

or

`BASIC /`*XX* `/BOOT` (Enter)

where *XX* is the mnemonic for the desired language (as listed previously for the BLPLANG environment variable). The first statement works if the measurement coprocessor has not yet been booted. If it has, use the second statement to force a re-boot. The language selection will take effect until the next re-boot.

For example, to select German, type:

`BASIC /GR /BOOT` (Enter)

The language specified (German in this case) will override the MS-DOS country code *and* BLPLANG until the next re-boot.

6

Display Systems and HP BASIC

The measurement coprocessor software (version C.00 and later versions) supports the use of several standard types of display adapters and monitors. The software produces a bit-mapped display on the CRT, very similar to that produced on a Series 300 display. Although various displays have different resolutions and aspect ratios, the software "remaps" the display to give a correct aspect ratio for vector graphics in each case. Also, the entire available display area is used, with no clipping.

Note

This chapter describes how HP BASIC uses the various supported PC display systems. HP BASIC provides several powerful graphics capabilities that allow you to create images on your display screen. If you have a color display, you can make use of color graphics to create especially impressive images. For information on HP BASIC graphics programming techniques, refer to chapter 10 in *Programming with HP BASIC*.

Supported Display Systems

In order to use the measurement coprocessor you will need to have one of the following supported types of display systems connected to your PC:

- Any VGA (Video Graphics Adapter) display system. Both color and monochrome VGA monitors are supported. For example, you could use an HP D1180A Video Graphics Adapter with either an HP D1182A VGA Color Display or an HP D1181A/G/W VGA Monochrome Display.

- Any EGA (Enhanced Graphics Adapter) display system. For example, you could use an HP 45983A Enhanced Graphics Adapter and an HP 35743A Enhanced Graphics Display.
- A Multimode monochrome display system consisting of the HP 45981A Multimode Video Adapter and HP 35731A Monochrome Monitor.
- Any Hercules-compatible display adapter and monitor.
- The HP 35732A Monochrome Plus Display Adapter and HP 35731D Monochrome Plus Display (automatically used in Hercules-compatible display mode).
- The AT&T 640 by 400 Monochrome display adapter and monitor.
- The Compaq Portable III flat-panel display.

Note

For optimum graphics performance, a VGA display system is highly recommended. The VGA display offers higher resolution and is the primary compatibility model for the measurement coprocessor software.

If you change your display to a different type after installing HP BASIC, you will need to run the INSTALL program again to load the correct display driver.

The Alpha Display

Let's begin by looking at the alpha (text) display generated by the measurement coprocessor.

Alpha Resolution

The alpha display is generated using a software font, or character set, which is provided by the measurement coprocessor software. This font has a 7-by-13 pixel character size within an 8-by-14 pixel cell. (A *pixel* is a single dot on the screen, the smallest element of an image on the display.) The same font is used for all display types. Thus, the number of lines of text on the screen depends on the display. The following table gives the resolution of each display and the number of rows and columns of alpha information that comprise a full screen.

Display Type	Pixel Resolution (width by height)	Number of Text Lines	Line Width (Characters)
VGA	640 by 480	34	80
EGA	640 by 350	25	80
HP Multimode	640 by 400	28	80
AT&T	640 by 400	28	80
Hercules	720 by 348	25	90
Compaq Portable III	640 by 400	28	80

For example, on a VGA monitor you can display up to 34 lines of text (including the softkey labels). Each line can be up to 80 characters wide. On an EGA monitor you can display up to 25 lines of text (including softkeys), each line being up to 80 characters wide.

Note

If you are writing an HP BASIC application program that may be used with more than one display type, remember to write your message screens so that they will fit on the different displays. For example, if you write your messages to fill the output area of a VGA display, the messages won't fit on an EGA display. To avoid the problem, leave 9 blank lines at the bottom of the output area on the VGA screen.

Display Characters

The measurement coprocessor software font consists of an HP Roman 8 character set equivalent to that used by HP 9000 Series 300 computers. This font is used for all alpha operations. Thus, the full HP Roman 8 extended-ASCII character set is supported by the measurement coprocessor. (This character set consists of the US ASCII standard characters plus the Roman 8 extended characters from 128 through 255. The Roman 8 extended characters are primarily used for localization to various European languages.)

Note The extended characters in the HP Roman 8 soft font differ from the extended characters in the PC-8 display font used in the MS-DOS environment. For a complete listing of the U.S./European display characters, refer to the "Useful Tables" appendix in your HP BASIC *Language Reference* manual. The measurement coprocessor uses the same U.S./European display characters as a Series 300 computer with an HP 98700 Display Controller.

The HP BASIC Bit-Mapped Graphics Display

The measurement coprocessor supports several display systems, each with its own resolution and screen aspect ratio. To do this, and yet fill the entire available display area, the process known as "bit-mapping" is used. The measurement coprocessor bit-mapped display is very similar to the Series 300 bit-mapped display.

In a bit mapped display system, bits in display memory are mapped to pixels on the display screen. The pixels are turned "on" or "off" at each location on the display screen according to the values of the corresponding bits in display memory. This allows the entire display area to be used without "clipping" of graphics images.

Aspect Ratios

The *aspect ratio* of a display screen is generally defined as its width divided by its height. However, there are two ways to measure the width and height of the screen. You can measure the physical width and height, or you can use the width and height in pixels (dots on the screen).

The *screen aspect ratio* is simply the physical width of the screen divided by the physical height. You could measure the width and height in inches or millimeters, but HP BASIC uses a standard unit for vector measurements called the GDU (Graphics Display Unit). The HP BASIC graphics screen on a VGA monitor has a width of approximately 133 GDUs and a height of 100 GDUs. Thus, the screen aspect ratio is approximately 1.33. When you are creating vector-based graphics, which is the normal method for HP BASIC, the screen aspect ratio is what you are interested in. The HP BASIC "RATIO" statement returns the screen aspect ratio for the current display. For further information about GDUs and the RATIO statement, refer to chapter 10 in *Programming with HP BASIC*.

The *raster aspect ratio* is simply the number of pixels in the horizontal direction divided by the number of pixels in the vertical direction. The VGA screen is 640 pixels wide and 480 pixels high. Thus, the raster aspect ratio is $640 \div 480$, or approximately 1.33.

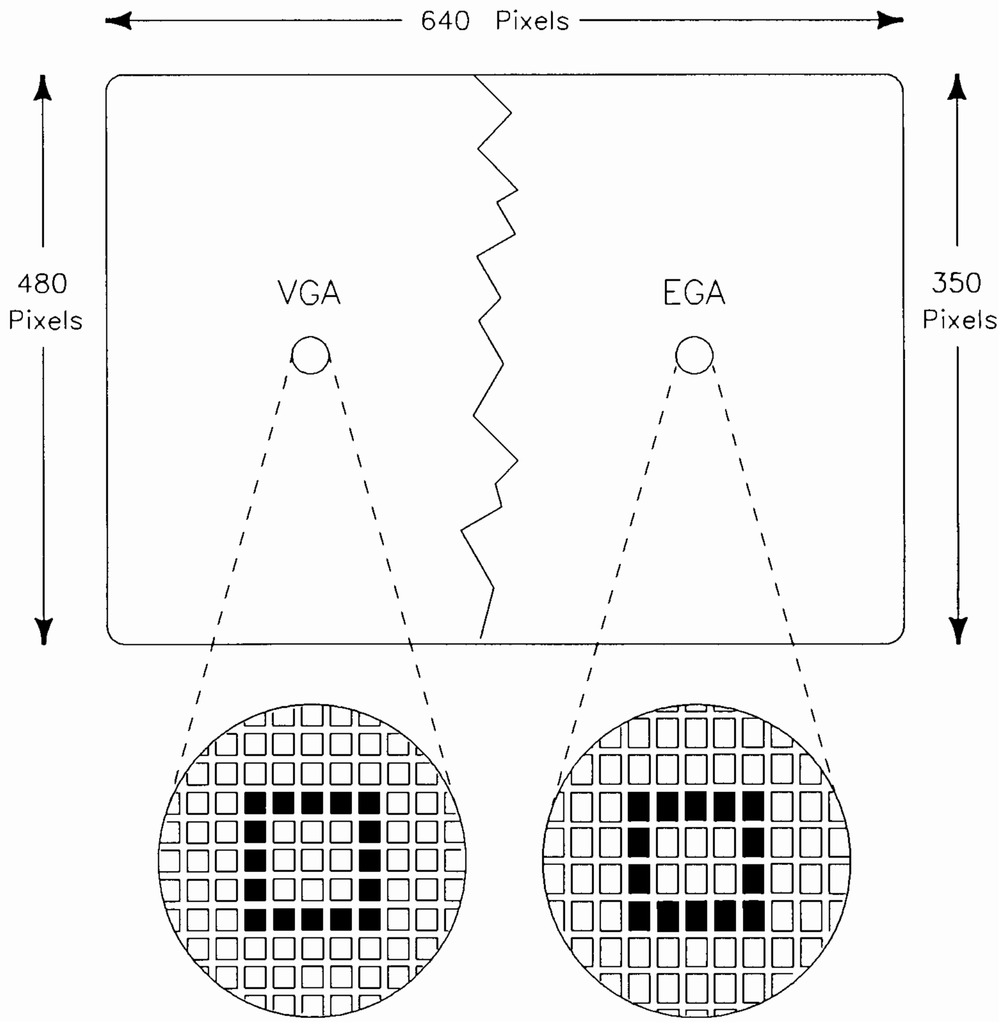
The reason that the screen aspect ratio and raster aspect ratio are the same in this case is that the VGA monitor has *square pixels*. That is, the pixels have equal heights and widths. This means that there is no problem mapping a vector image to the VGA screen, and the aspect ratio will be preserved in a raster dump of the screen to a printer. (Assuming that your printer has square pixels.)

Note The rest of this section covers the aspect ratios of non-VGA display systems and some special considerations concerning these displays. If you are using a VGA display system, you need not be concerned with the rest of this discussion concerning aspect ratios.

The EGA monitor also has a screen aspect ratio of about 1.33. The EGA monitor has a screen width of 640 pixels, but its height is only 350 pixels. Thus the raster aspect ratio would be $640 \div 350$, or about 1.83 for an EGA monitor. To avoid aspect ratio problems with vector graphics, the measurement coprocessor software "remaps" the EGA display to approximate a 480 pixel height.

How does this work? Quite simply, when you (or your program) execute an HP BASIC graphics command such as DRAW, the software calculates the physical positions of the pixels to turn on, and lights the pixels closest to those positions. An example is very much in order, since this can be confusing. The illustration on the following page shows an HP BASIC screen. However, the left half of this screen is shown as a VGA screen and the right half is shown as an EGA screen. Both the VGA and the EGA screens are 640 pixels wide. However, the EGA screen has fewer pixels in height (350) than the VGA (480). Thus, the EGA pixels are taller, but have the same width as the VGA pixels.

In the illustration, the small square on the left (VGA) side is five pixels wide and five pixels high. This same square is represented on the right (EGA) side using five pixels of width, but only four pixels of height. The physical size is approximately the same because the pixels are taller.



In effect, the EGA screen has been *rescaled* to have an effective height of 480 pixels. Thus, you can draw the same images on the EGA screen as on the VGA screen, or at least a close approximation. However, the EGA images will have lower resolution in the vertical direction.

We have discussed the EGA display as an example, but the same discussion will also apply to all of the supported displays except the VGA display and the Compaq Portable III panel display, which have square pixels.

The following table gives graphics parameters for all of the supported displays:

Display Type	Pixel_ratio	Width (pixels)	Height (pixels)	Scaled Height (pixels)	RATIO
VGA	1.000	640	480	480	1.334029
EGA	0.729	640	350	480	1.334029
HP Multimode	0.800	640	400	500	1.280561
AT&T	0.800	640	400	500	1.280561
Hercules	0.671	720	348	518	1.390716
Compaq Portable III	1.000	640	400	400	1.601504

The "Pixel_ratio" parameter is the width of an *individual pixel* divided by its height. If "Pixel_ratio" is 1.000, the pixel is square. This parameter is used to rescale the height of the screen.

The "Width" and "Height" of the display in pixels are the normal, published parameters for each display.

The "Scaled Height" parameter is the *effective* height in pixels of the rescaled HP BASIC display. This value is equal to "Height" divided by the "Pixel_ratio" parameter.

The value of "RATIO" is equal to the screen aspect ratio of the display, and is returned by the "RATIO" statement. You can calculate this parameter as follows:

```
RATIO = (Width – 1) ÷ (INT (Height ÷ Pixel_ratio) – 1)
```

Note

Because the measurement coprocessor software rescales each display as though it has square pixels, you should encounter no problems creating graphics on the screen. Most HP BASIC graphics commands are vector based. However, if you attempt to dump a raster image of your CRT to a printer, you may encounter aspect ratio problems. Refer to "Dumping the Graphics Screen" later in this chapter.

Combined Mode and Separate Mode

The measurement coprocessor and the HP 9000 Series 300 computers both default to "combined alpha and graphics" mode. In combined mode, the alpha and graphics screens are merged into one, inseparable display. The commands GRAPHICS ON, GRAPHICS OFF, ALPHA ON, and ALPHA OFF have no effect in combined mode.

If you want to enter the "separate alpha and graphics" mode, execute the following HP BASIC statement:

```
SEPARATE ALPHA FROM GRAPHICS
```

Note

Separate mode is supported only for color monitors (VGA or EGA) and for the VGA monochrome monitor. You cannot use separate mode with other monochrome monitors. If you try to enter separate mode with a monitor that doesn't support that mode, an error will result.

In separate mode, you can turn the alpha screen and graphics screens on and off separately. You can use the GRAPHICS ON, GRAPHICS OFF, ALPHA ON, and ALPHA OFF commands, or you can press the "Graphics" and "Alpha" keys. For a complete discussion of separate and combined mode, refer to chapter 10 in *Programming with HP BASIC*.

To return to combined mode, execute:

```
MERGE ALPHA WITH GRAPHICS
```

The Graphics Area

The *graphics area* is the area on the CRT available for drawing graphics. The measurement coprocessor, like the HP 9000 Series 300 computers, uses the full area of the CRT for graphics. There is no “clipping” of graphics images. However, you can use the VIEWPORT statement to reduce the area on the screen within which graphics are drawn.

Let's consider what happens when you execute the following BASIC statements.

```
GINIT
GCLEAR
FRAME
```

- The GINIT statement initializes several graphics parameters, including the limits of the graphics area. After the GINIT statement, the graphics area uses the entire area of the display screen.
- The GCLEAR statement clears the graphics screen.
- The FRAME statement draws the outline of the currently defined graphics area, which in this case encompasses the entire display screen.

Note

The limits set by GINIT are equivalent to the statement:

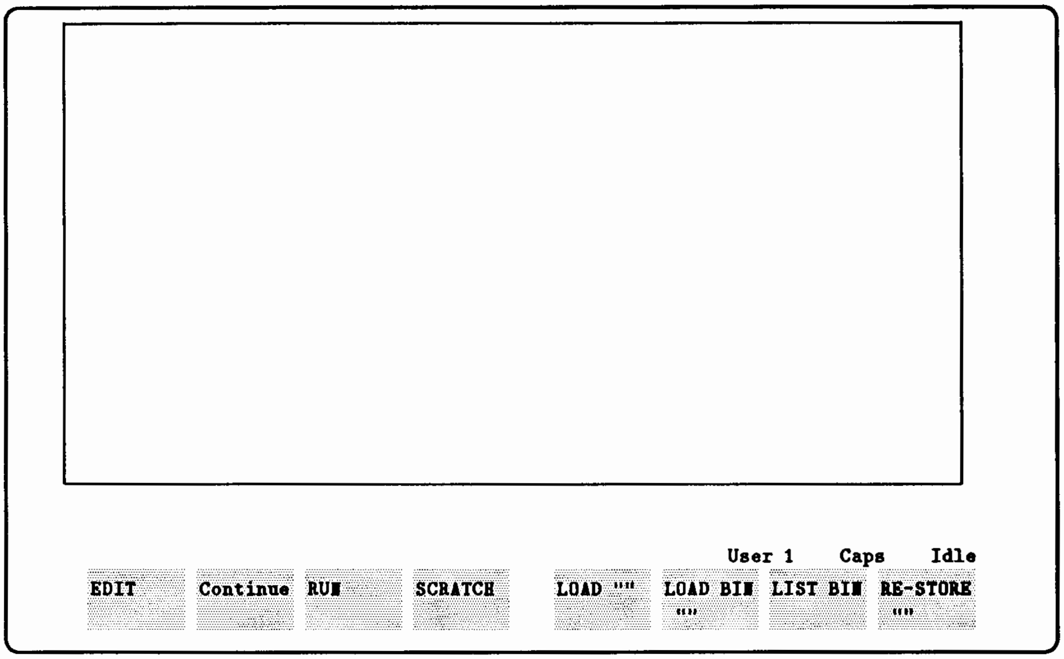
```
VIEWPORT 0,RATIO*100,0,100
```

That is, the x-minimum and y-minimum are both 0, the x-maximum is RATIO*100, and the y-maximum is 100. (The RATIO function returns the aspect ratio of your CRT monitor. For further information on using the RATIO and VIEWPORT statements, refer to chapter 10 in *Programming with HP BASIC* and to the HP BASIC *Language Reference* manual.)

There is a problem with using the entire screen area for graphics. The command line and softkey labels at the bottom of the screen will interfere with the graphics image. Although you can turn off the softkey labels with the “Menu” key, you may want to avoid drawing in the area occupied in the menu and command line area. This is especially true in “combined” mode, since whenever you execute a command you will erase part of the lower section of the graphics image. (This is true for Series 300 computers also.) To avoid this problem, execute the following BASIC statement:

```
VIEWPORT 0,RATIO*100,20,100
```

Now execute the FRAME statement. The y-minimum parameter in the VIEWPORT statement is "20", which raises the bottom of the graphics area above the command line, as shown in the following screen.



If you are using isotropic units, you'll need to reduce the width as well as the height of the graphics area to preserve the correct aspect ratio. For example, you could use the statement

```
VIEWPORT RATIO*10,RATIO*90,20,100
```

which preserves the aspect ratio and centers the graphics area within the width of the screen.

Dumping the Graphics Screen

If you have a VGA display adapter and monitor, you can do a raster dump of the graphics screen to a printer without any additional complications (provided your printer has been configured). Just specify the address of the printer in a `DUMP DEVICE IS` statement, then press the "Dump Graphics" key or execute:

```
DUMP GRAPHICS  [Enter]
```

from HP BASIC. A raster dump of the graphics screen will occur pixel-by-pixel, just as for an HP 9000 Series 300 system. (In combined alpha/graphics mode the entire screen is dumped — both alpha and graphics information is output to the printer in raster form. In separate mode, only graphics information is output.)

Note

Refer to chapter 9, "Using Printers and Plotters with the Measurement Coprocessor," for further information about using HP-IB, serial, and parallel printers.

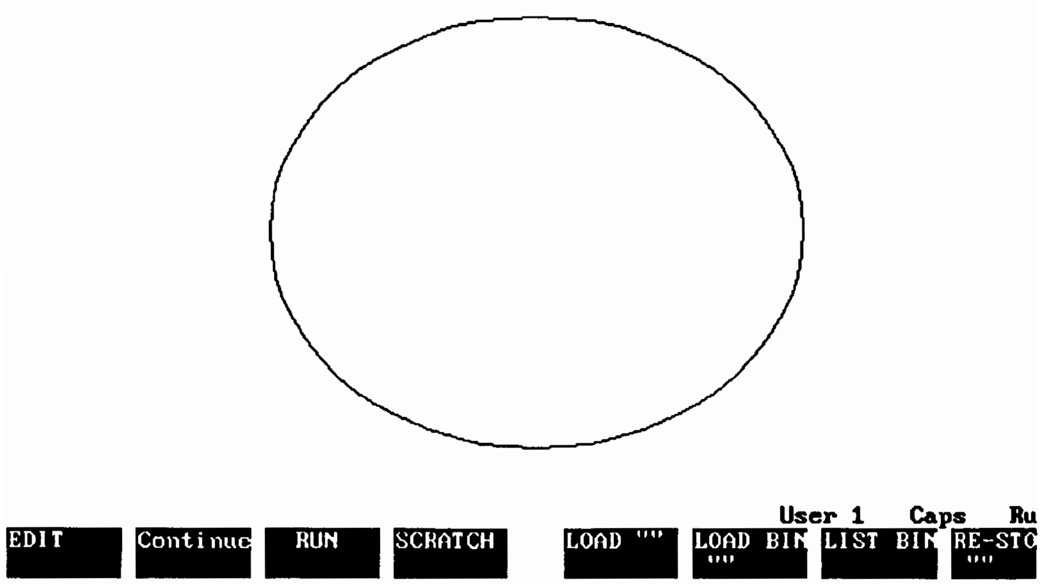
This same method will work for EGA and other supported displays, but there is a complication. As mentioned earlier, the graphics screen has been *rescaled* to support *vector* graphics with the correct aspect ratio even though a monitor may not have square pixels.

Let's look at an example of vector graphics on an EGA display system. When you use the POLYGON statement to draw a circle, you are using vector graphics. For an EGA monitor, BASIC automatically rescales the graphics display and draws an image that is more *pixels* wide than high. The calculation is such that the *actual* width in inches is the same as the height, producing a circle, rather than an ellipse.

The following program will draw a circle on the CRT screen and will then do a raster dump of the screen:

```
10   GINIT                  ! Initialize graphics parameters.
20   CLEAR SCREEN
30   !
40   MOVE RATIO*50,50       ! Move to center of screen.
50   POLYGON 30             ! Draw circle of radius 30.
60   !
70   DUMP DEVICE IS 9       ! Dump to serial printer.
80   DUMP GRAPHICS
90   END
```

If you run this program with a VGA display system (or the Compaq Portable III panel display), the circle on the screen will be dumped as a circle on the printout. However, if you are using an EGA display (or other display with non-square pixels), the circle on the screen will appear as an ellipse in the dump:



The solution is to replot the circle so that it has an equal number of pixels in each direction. In the previous program the GINIT statement set the aspect ratio to be equivalent to:

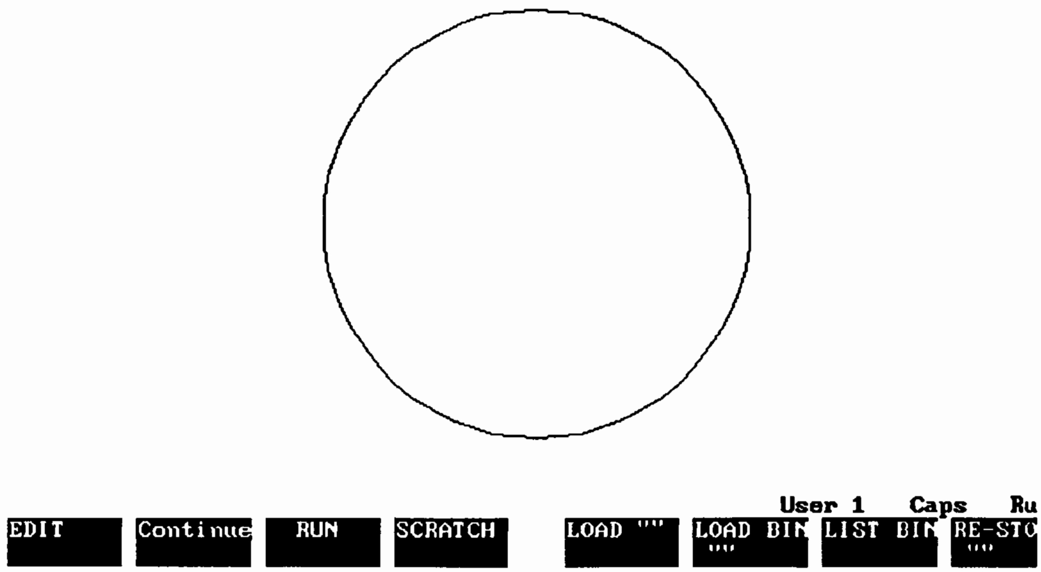
```
WINDOW 0,RATIO*100,0,100
```

where RATIO is the calculated ratio of the screen width divided by the screen height. We need to replace the screen aspect ratio given by the RATIO function with the raster aspect ratio for the display (EGA in this case).

You can calculate the raster aspect ratio by using the GESCAPE function. The following program uses GESCAPE to return values to an array, and then calculates the raster ratio based on the number of horizontal pixels, Ge(2), and the number of vertical pixels, Ge(3).

```
10   GINIT                      ! Initialize graphics parameters.
20   CLEAR SCREEN
30   !
40   OPTION BASE 0
50   INTEGER Ge(7)              ! Dimensions integer array.
60   GESCAPE CRT,3;Ge(*)        ! Returns CRT parameters to array.
70   WINDOW 0,Ge(2)/Ge(3)*100,0,100       ! Rescales for dump.
80   !
90   MOVE Ge(2)/Ge(3)*50,50          ! Move to center of screen.
100  POLYGON 30                      ! Draw circle of radius 30.
110  !
120  DUMP DEVICE IS 9                ! Dump to serial printer.
130  DUMP GRAPHICS
140  END
```

This program will dump a circle regardless of the type of display in use. However, for an EGA or other display with non-square pixels, the image on the CRT will be an upright ellipse. In other words, we've calculated the correct aspect ratio for the dump, even though the ratio is wrong for the CRT. The dumped image will look like the following figure.



Gray Scale Mapping on VGA Monochrome Displays

The measurement coprocessor has two modes of operation for VGA monochrome monitors:

- In the default mode (for pens 0 through 7), the monitor functions as a simple monochrome display, with no shades of gray.
- In the *NTSC gray scale* mode, HP BASIC will treat the display as if it was a color display, but will map the colors to shades of gray according to the NTSC (National Television Standards Committee) standard. The PEN values are mapped to the same monochrome intensities whether you are in "non-color mapped" mode or "color mapped" mode.

The following table lists the intensities for both modes:

PEN	VGA Monochrome Intensity Default Mode (percent)	 NTSC Mode (percent)	Corresponding Color
0	0	0	(Black)
1	100	100	(White)
2	0	30	(Red)
3	100	89	(Yellow)
4	100	59	(Green)
5	100	70	(Cyan)
6	0	11	(Blue)
7	0	41	(Magenta)

If you have a VGA monochrome monitor and you want to use the default mode, you don't have to do anything. The measurement coprocessor is configured in default mode automatically. If you want to use NTSC gray scale mapping, you will have to enable NTSC mapping as follows:

1. The configuration file named "NTSC.CON" is found on your "Manual Examples, LIFINIT, and Selected CSUBs" disk. Insert this disk in drive A:.
2. Execute the following MS-DOS command:

   ```
   COPY A:\NTSC.CON C:\BLP\BLP.CON
   ```
   Enter

   This will overwrite any modifications to your configuration that you have stored in "BLP.CON". You will have to make those modifications in the new "BLP.CON" file.

Note

If you have made extensive modifications to "BLP.CON" (for example, you have configured disk drives and set up "keydefs"), you may not want to overwrite your "BLP.CON" file. To avoid this, you can modify "BLP.CON" manually using an MS-DOS text editor:

To enable NTSC mapping, add the following line at the end of the "BLP.CON" file (type the line exactly as shown, with spaces):

```
6 1 0
```

To disable NTSC mapping, remove the line from "BLP.CON"

Make sure that you make no other modifications to the file with your MS-DOS editor, and that you store the file as an unformatted ASCII file.

7

File Systems and Mass Storage

This chapter describes the mass storage systems supported by the measurement coprocessor, and how to access data and program files from BASIC. *For information on sharing data and program files with MS-DOS applications, refer to chapter 10.*

First, let's look at some general concepts.

Mass Storage Concepts

As the term "mass" suggests, mass storage devices are designed to store large quantities of data. Just how much data constitutes a large amount depends on the device itself. Most mass storage devices are capable of storing on the order of hundreds of thousands to several million items.

Mass storage devices are also called "secondary" storage media, the "primary" storage being computer memory (Random Access Memory, or RAM). Because of this secondary nature, they are not generally expected to be as fast in accessing their data as primary storage (RAM). Although mass storage can be accomplished using various kinds of media, usually some form of magnetic media — disk or tape — is used.

Besides having the ability to store data, mass storage devices are capable of providing means for keeping data organized so that logical groups may be accessed systematically and efficiently. The information on a mass storage device is organized into volumes and files. In some cases, volumes are divided into collections of files known as directories.

Files, Volumes, and Directories

A **file** is a collection of information stored on a disk or other mass storage medium and identified by a **file name**. The file may be a simple collection of ASCII bytes, or it may include a header or other formatting information.

A **volume** is merely a collection of files. Normally, a flexible disk will be one volume. A hard disk may be partitioned into more than one volume. Each volume or mass storage medium may contain several files.

Some mass storage systems have volumes with hierarchical organizations. In such systems, each volume can be subdivided into several **directories**, each of which can contain several files. In fact, each directory can be further divided into subdirectories. The MS-DOS operating system uses a hierarchical directory structure.

File Systems

In order to provide the highest possible mass storage performance, but also provide compatibility with other hardware and software systems, the measurement coprocessor provides three different **file systems**:

- The **DFS** (DOS File System) binary provides direct access from HP BASIC to the MS-DOS file system. *DFS is the primary file system for the measurement coprocessor.* It provides the fastest file access and the best compatibility with MS-DOS and MS-DOS applications. Normally, you will want to use DFS file access in your programs.
- The **LIF** (Logical Interchange Format) file system is supported by the measurement coprocessor to provide compatibility with HP 9000 Series 200 and Series 300 computers.
- The **HPW** (virtual-LIF) file system is supported to provide full compatibility with previous versions of the measurement coprocessor (the HP 82300A/B BASIC Language Processor).

In addition, the measurement coprocessor supports two optional mass storage systems:

- The **HFS** (Hierarchical File System) binary is available as a separate product. This binary provides the HP 9000 Series 300 Hierarchical File System for the measurement coprocessor. HFS is described in detail in the *Getting Started* manual that comes with the HP 82313A Hierarchical File System.
- The **SRM** (Shared Resource Management) file system is used by the measurement coprocessor with the optional HP 50963A SRM Interface Card. The SRM file system is described in appendix A, "Using SRM with the Measurement Coprocessor."

Note

There are four main file types in HP BASIC: PROG, BDAT, ASCII, and *untyped* DOS or HP-UX files. Untyped data files are of the DOS file type for the DFS file system. For the other file systems, untyped data files are of the HP-UX file type. Otherwise, file type is independent of the file system in use. These file types are described in detail in *Programming with HP BASIC* (chapter 6, "Data Storage and Retrieval").

Using the DOS File System (DFS) from HP BASIC

The DFS (DOS File System) binary allows you to access the MS-DOS file system directly from HP BASIC. This binary is installed as part of the HP BASIC installation process. The DFS binary provides file and directory access similar to that provided by the HFS binary and provides significantly faster mass storage access than the LIF and HPW file systems. An additional advantage is that DFS files can easily be shared with MS-DOS applications. (Refer to chapter 10 for more information.) *You should use DFS file access in your programs for most applications.*

Note that the DFS binary only gives you access to the MS-DOS disk drives of your HP Vectra PC or other personal computer. External LIF disk drives may be connected to the measurement coprocessor through the HP-IB interface, but you will have to use the LIF (or optional HFS) file system to access them.

If you have files stored in another file system (LIF, HPW, HFS, or SRM), you can use the HP BASIC "COPY" command to make DFS copies of them. (Refer to "Copying Files Between File Systems" later in this chapter.) Also, a utility program named "CATCOPY" is provided that allows you to batch copy files from LIF or HPW volumes to a DFS directory. *In either case, the copying process automatically translates the file format.*

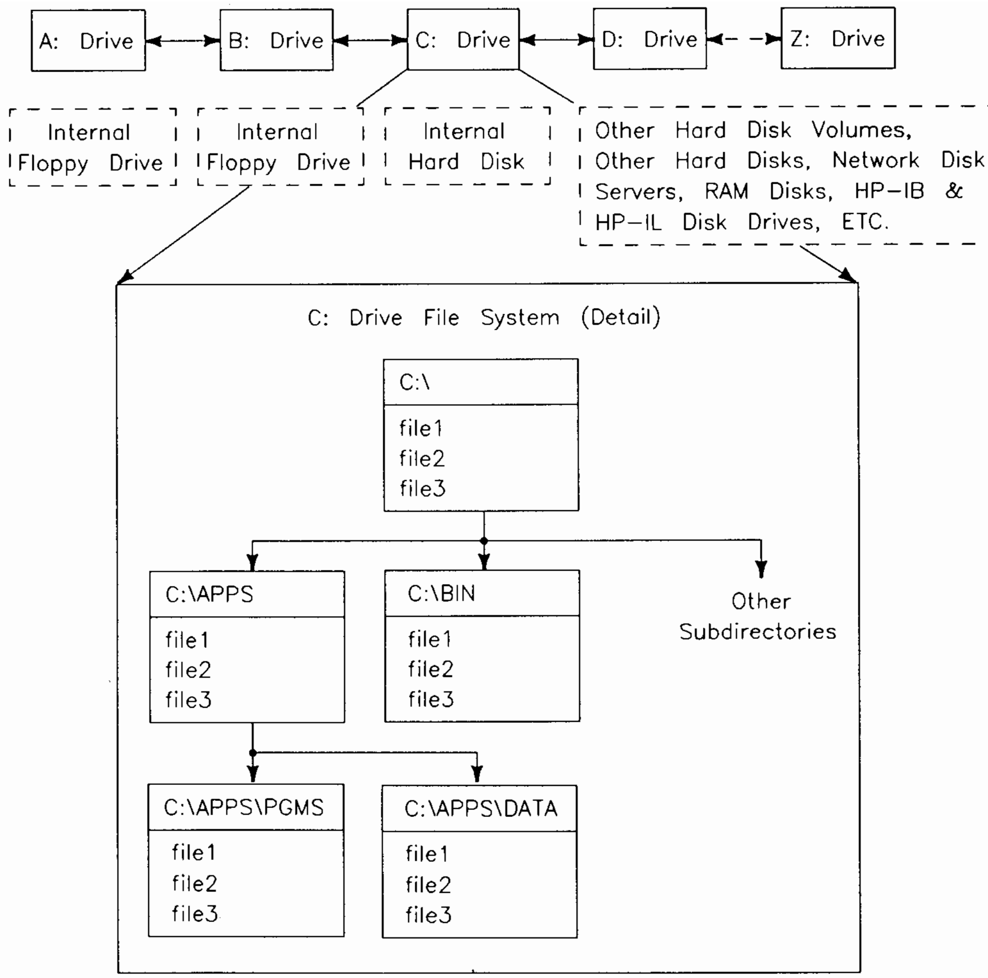
Note

If you have programs written for the HP 9000 Series 200 or Series 300 computers, or for earlier versions of the measurement coprocessor, you may need to modify these programs to work with DFS mass storage. Refer to the *Porting Guide for the Measurement Coprocessor* for specific instructions. The CATCOPY utility is also described in that manual.

Let's begin by looking at the MS-DOS hierarchical file system itself.

An Overview of the MS-DOS File System

Most MS-DOS-based personal computers usually do not use external drives. They use built-in flexible disk and hard disk drives. A typical PC uses one or two flexible disk drives and a single hard disk drive. A hard disk drive may be partitioned into multiple volumes. The following illustration shows the MS-DOS volume and directory structure for a typical system.



Note that each flexible disk drive ("A:" and "B:") constitutes one volume. The internal hard disk drive may be partitioned into multiple volumes (for example, "C:" and "D:"). Other mass storage volumes (through "Z:") may include hard disks, network disk servers, RAM disk volumes, and external disk drives.

Any mass storage volume ("A:" through "Z:") may be subdivided into a hierarchy of directories and subdirectories. In the illustration, "C:" is shown to be subdivided into a root directory ("C:\") and several subdirectories ("C:\APPS", "C:\APPS\PGMS", "C:\APPS\DATA", etc.)

As mentioned in chapter 1, all you have to do to select a particular drive or volume in MS-DOS is enter its drive/volume ID. For example:

```
A: (Enter)
```

selects flexible disk drive "A:", while:

```
C: (Enter)
```

selects hard disk volume "C:".

Within "C:" you can change directories with the change directory command. For example:

```
CD C:\APPS (Enter)
```

changes the directory to "C:\APPS", while:

```
CD C:\ (Enter)
```

returns to the root directory.

The advantage of the hierarchical file system is that it makes it much easier for users to organize their files. Files can be stored in subdirectories appropriate to their functions. For example, the *root* directory of hard disk volume "C:" is identified as:

```
C:\
```

In the example shown in the figure on the previous page, two directories have been created under the root directory, one (named "APPS") to store applications programs, the other (named "BIN") to store various system programs (binaries). These are identified with the path names:

```
C:\APPS
C:\BIN
```

Two subdirectories have been created under the "APPS" directory, one to store auxiliary programs ("PGMS") and another to store data files ("DATA"). These are identified with the path names:

```
C:\APPS\PGMS
C:\APPS\DATA
```

The **path name** of an MS-DOS directory starts with the root "\" and includes the name of each subdirectory, with the back slash ("\") used as a delimiter.

You can also use a path name to specify a file in a directory. For example, the file named "DFILE" in the directory "DATA" has the following path name:

```
C:\APPS\DATA\DFILE
```

This path name uniquely describes the file from any directory within the MS-DOS file system.

For further information about using MS-DOS, refer to the reference documentation supplied with your MS-DOS operating system.

Formatting a DFS Disk

Before you can use a mass storage medium such as a new flexible disk, you must **format** it. (You may also want to format a previously used disk in order to remove all of the information on it. *Just make sure that you don't need any of the files on the disk before you format it.*)

Note HP BASIC provides an "INITIALIZE" statement, which is used for the LIF file system. INITIALIZE is not implemented by the DFS binary. To prepare disks for DFS use, format them from MS-DOS using the "FORMAT" command.

Formatting puts the necessary "sector" and "track" identification on the disk so that the operating system can store and retrieve files. To format a flexible disk in drive "A:", type:

```
FORMAT A: (Enter)
```

from the MS-DOS command line. Insert the disk to be formatted and press (Enter) again when prompted.

If you are in HP BASIC, you can use OUTPUT 19 (the MS-DOS communications port) to execute an MS-DOS "FORMAT" command. For example:

```
OUTPUT 19; "FORMAT A:"  [Enter]
```

from the HP BASIC command line is equivalent to the previous statement on the MS-DOS command line. Refer to chapter 4 for further information about OUTPUT 19.

Caution

Initializing and formatting a hard disk is a "precarious" operation since the hard disk contains so much information. *The initialization and formatting process destroys all data on the hard disk and normally is only done during the initial setup of your computer.* Follow the instructions in your computer owner's documentation very carefully.

Once you have formatted an MS-DOS disk volume (flexible disk or hard disk), you can create directories and store files in that volume from within MS-DOS. *In addition, you can access files and directories in that volume directly from HP BASIC through DFS.* The following section tells how to access the DOS file system from HP BASIC.

Accessing DFS Files and Directories

Now let's look at how you can access MS-DOS files and directories from HP BASIC using the DFS binary. The statements and syntax for HP BASIC file access are somewhat different from MS-DOS, but you can perform essentially the same functions.

Naming Files and Directories

Each file name in the DFS file system can consist of up to eight alphanumeric characters followed by a "dot extension" of up to three alphanumeric characters. All alpha characters are "case-folded" into uppercase characters. The dot extension, if included, must follow a period. However, the dot extension (and the period) are optional.

Each subdirectory name is actually a file name, so the same naming rules apply.

Specifying Directories, Files, and Volumes

As mentioned previously, the names of MS-DOS volumes, directories, and files are combined to form a complete path name, which uniquely specifies a particular file location in the file system. The back slash ("\") is used as a delimiter. However, there is one difference between DFS and MS-DOS path names. The following are valid for MS-DOS:

```
C:\DIRECTRY.ONE\DATAFILE.ONE
C:\DIRECTRY\SUBDIR_1\DATAFILE
```

The corresponding paths for HP BASIC are:

```
\DIRECTRY.ONE\DATAFILE.ONE:DOS,C
\DIRECTRY\SUBDIR_1\DATAFILE:DOS,C
```

Notice that for HP BASIC the volume specifier ":DOS,C" follows the path and file name, while for MS-DOS, the volume specifier "C:" precedes the path. The volume specifier ":DOS,C" is called a Mass Storage Volume Specifier, or MSVS.

In HP BASIC, as in MS-DOS, the full path name always starts at the root "\".

The DFS binary provides another feature that differs from MS-DOS. For your convenience, you may use either the back slash ("\", following the MS-DOS convention) or the forward slash ("/", following the convention used by the HFS, HP-UX, and SRM systems) to separate path names. Thus:

```
/DIRECTRY/SUBDIR_1/DATAFILE:DOS,C
```

is equivalent to:

```
\DIRECTRY\SUBDIR_1\DATAFILE:DOS,C
```

The Current Mass Storage

When you boot HP BASIC, the **current mass storage** becomes the directory from which you booted BASIC. For example, if you boot BASIC from the MS-DOS directory "C:\APPS", the current mass storage will be that directory, or "\APPS:DOS,C". The current mass storage will remain in effect until you either reboot, or execute a "MASS STORAGE IS" (or "MSI") statement. If you exit BASIC and re-enter without rebooting, the current mass storage will be the same as it was before you exited.

Note

The current mass storage is maintained separately from the current MS-DOS directory. Exiting BASIC and changing the directory will have no effect on the current mass storage on re-entry to BASIC (unless you reboot the measurement coprocessor).

Using the MASS STORAGE IS Statement

The "MASS STORAGE IS" statement is used in HP BASIC to select the mass storage volume (or drive) and directory that you want to access. This statement can specify just the volume or drive. For example:

```
MASS STORAGE IS ":DOS,A" (Enter)
```

makes flexible disk drive "A:" the current mass storage. Using the short form, "MSI":

```
MSI ":DOS,C" (Enter)
```

makes hard disk volume "C:" the current mass storage.

You can also specify an individual directory by its path name. For example, if hard disk volume "C:" is the current mass storage:

```
MSI "\BLP\PROGS" (Enter)
```

will make "C:\BLP\PROGS" the current directory. The more complete statement is:

```
MSI "\BLP\PROGS:DOS,C" (Enter)
```

which explicitly specifies both volume and directory.

You can always specify a directory using its complete path name (for example, "\BLP\PROGS". However, if you are in the directory immediately superior or "above" the desired directory, you don't need to specify the whole path. For example, if the current directory is "\BLP", you can use the following statement to change the directory to "\BLP\PROGS":

```
MSI "PROGS"  [Enter]
```

You can move from a subordinate directory to the one immediately above it using the ".." convention. For example, to move from "\BLP\PROGS" back to "\BLP", type:

```
MSI ".." [Enter]
```

If you want to change to a directory that is more than one level away, you will have to specify the whole path. If the directory is on a different volume, you will also have to specify the volume. Thus, if the current mass storage volume is ":DOS,C", you can use:

```
MSI "\BLP\PROGS" [Enter]
```

to change directly to the directory "\BLP\PROGS". On the other hand, if the current mass storage is ":DOS,A", use the following instead:

```
MSI "\BLP\PROGS:DOS,C"  [Enter]
```

The following statement is equivalent to the last.

```
MSI "/BLP/PROGS:DOS,C"  [Enter]
```

Note

In any HP BASIC statement that contains a DFS path name, you can use either the forward slash "/" or the back slash "\" as a separator. However, for consistency we'll use the back slash "\" throughout the rest of this chapter.

Listing a Catalog

The HP BASIC "CAT" command lists a directory somewhat like the MS-DOS "DIR" command. However, CAT gives you more information. If you boot HP BASIC from the directory "C:\BLP", wait until the BASIC screen appears, and type:

CAT (Enter)

a catalog of "C:\BLP" will be listed. The catalog listing will be similar to that shown below:

Note

The simple form of the CAT statement (just "CAT" followed by (Enter)) is very useful when you want to determine which directory is the current directory.

```
DIRECTORY: C:\BLP
LABEL: HARD_DISK_C
FORMAT: DOS
AVAILABLE SPACE:     66952
                 FILE      NUM   REC      MODIFIED
FILE NAME        TYPE     RECS   LEN     DATE      TIME PERMISSION
============  ====== ======= ===== ========= ===== ==========
ACTDISP         DOS      18306     1   9-Mar-89 14:59  RW-RW-RW-
BASIC.EXE       DOS      72038     1   9-Mar-89 15:00  RWXRWXRWX
DLIFONT         DOS       3592     1   9-Mar-89 15:00  RW-RW-RW-
HPW.MSG         DOS      10733     1  21-Feb-89 11:16  RW-RW-RW-
HPW.OVR         DOS      49206     1  21-Feb-89 11:16  RW-RW-RW-
B1.EXE          DOS      71887     1   9-Mar-89 15:00  RWXRWXRWX
B2.EXE          DOS      51299     1   9-Mar-89 15:00  RWXRWXRWX
B3.EXE          DOS     123924     1   9-Mar-89 15:01  RWXRWXRWX
BLP.COM         DOS        126     1   4-Apr-89  9:57  RW-RW-RW-
SYSBA5          SYSTM     2410   256   9-Mar-89 15:03  RW-RW-RW-
BLP.MSG         DOS      10976     1   9-Mar-89 15:00  RW-RW-RW-
CONF.EXE        DOS      37263     1   7-Mar-89 22:54  RWXRWXRWX
CATCOPY         PROG        96   256   5-Mar-89 15:13  RW-RW-RW-
PROGS           DIR          0     1  21-Mar-89 14:29  RWXRWXRWX
```

The header at the top of the listing identifies the file system ("DOS"), the current directory ("C:\BLP"), and the volume label of the hard disk ("HARD_DISK_C"). The available space on the hard disk volume is given as the number of 256-byte records for which there is room. (In this case there is room for 66,952 256-byte records, or 17,139,712 bytes.)

The header is followed by a listing of the files in the directory. The first two columns give the file name and file type for each file. The file "SYSBA5" is the BASIC "SYSTEM" file. It, and most of the "DOS" type files in this directory make up the installed HP BASIC system. In addition, one "PROG" type file (the "CATCOPY" program) and one "DIR" or subdirectory ("PROGS") is shown. (The subdirectory "PROGS" is immediately subordinate to "C:\BLP" in the hierarchy. Thus, its full path name is "C:\BLP\PROGS".) Other file types include "BDAT" and "ASCII" files (not shown).

The third and fourth columns give the number of records and record length for each file. In general all files have a record length of 256 bytes, except for untyped DOS and HP-UX files, which have a record length of 1 byte.

The next two columns give the date and time of last modification for each file, and the last column contains the "permission" field. Most files are have the field "`RW-RW-RW-`" indicating read/write access. If you set the MS-DOS READ-ONLY attribute for a file, its field will become "`R--R--R--`". Executable files and directories have the field "`RWXRWXRWX`".

If the catalog listing is too long to fit in the screen, the listing will "scroll" off the top of the screen. You can use the cursor-control keys (the "arrow" keys) to scroll the listing up and down.

You can obtain a CAT listing of "C:\BLP" from another subdirectory by typing:

```
CAT "\BLP:DOS,C" (Enter)
```

or simply:

```
CAT "\BLP" (Enter)
```

The same rules apply to the CAT statement as to MASS STORAGE IS. If you are in the directory immediately superior ("above") the desired directory, you don't need to specify the whole path. For example, if the current directory is "\BLP", you can use the following statement to obtain a CAT of the directory "\BLP\PROGS":

```
CAT "PROGS"  Enter
```

You obtain a CAT of a directory from the one immediately subordinate to it ("under" it) using the ".." convention. For example, if "\BLP\PROGS" is the current directory, you can obtain a CAT of "\BLP", with:

```
CAT ".."  Enter
```

If you want to obtain a CAT listing of a directory that is more than one level away, you will have to specify the whole path in the CAT statement. (You could also change the directory to the one you want to list, of course.) If the directory is on a different volume, you will also have to specify the volume. Thus, if the current mass storage volume is ":DOS,C", use:

```
CAT "\BLP\PROGS"  Enter
```

On the other hand, if the current mass storage is ":DOS,A", use the following instead:

```
CAT "\BLP\PROGS:DOS,C"  Enter
```

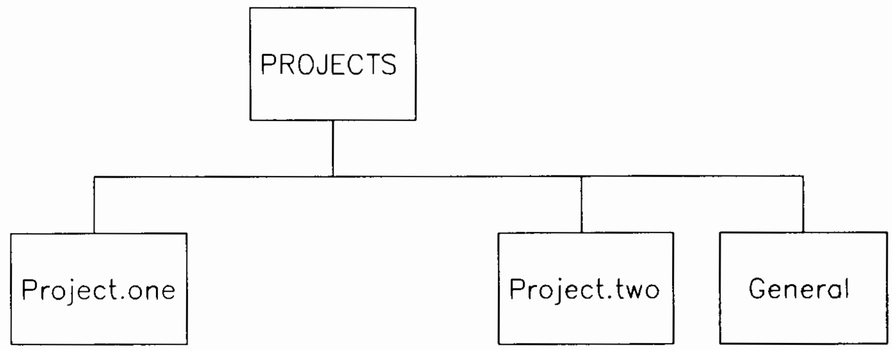
Creating DFS Directories

You can create DFS directories using the "CREATE DIR" statement. (This HP BASIC statement corresponds to the MS-DOS "MKDIR" statement.) One of the main advantages of a hierarchical directory structure is that you can keep your data and program files separated from your system files. Since your system files are all in the directory "C:\BLP", you will probably want to create a different directory for your own files. You could create a subdirectory under "C:\BLP". For example, the HP BASIC statement:

```
CREATE DIR "\BLP\PROGS:DOS,C"  Enter
```

was used to create the directory "C:\BLP\PROGS", mentioned previously.

On the other hand, you can create an entirely new directory structure starting at the DOS root directory. As an example, let's look at how to create the directory structure shown below:



First, we'll create the directory "\PROJECTS" under the root:

```
CREATE DIR "\PROJECTS:DOS,C"   Enter
```

The statement:

```
CREATE DIR "\PROJECTS"   Enter
```

would do the same provided ":DOS,C" is the current mass storage volume.

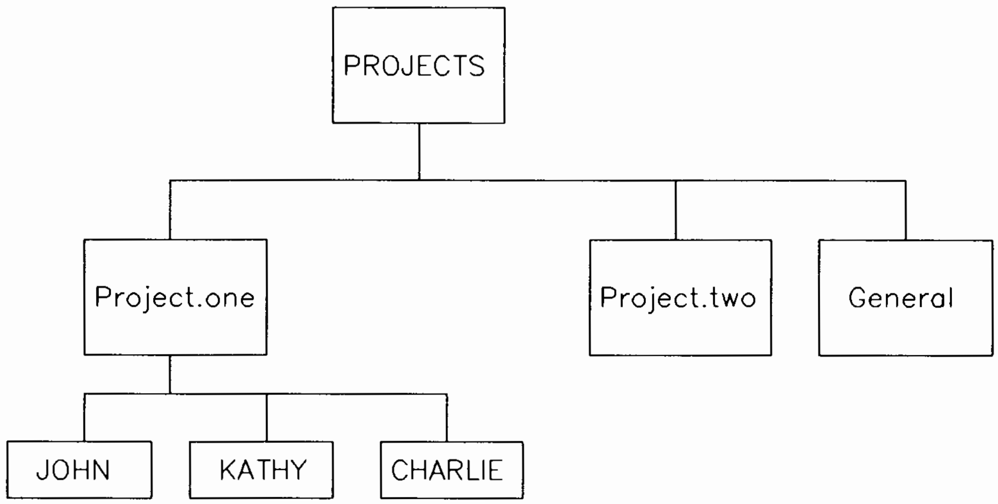
Now let's create the three subdirectories shown in the figure:

```
MSI ":DOS,C"   Enter
CREATE DIR "\PROJECTS\PROJECT.ONE"   Enter
CREATE DIR "\PROJECTS\PROJECT.TWO"   Enter
CREATE DIR "\PROJECTS\General"   Enter
```

To add another level to the hierarchy, we'll create three three subdirectories under "\PROJECTS\PROJECT.ONE":

```
MSI "\PROJECTS\PROJECT.ONE"  Enter
CREATE DIR "KATHY"  Enter
CREATE DIR "JOHN"  Enter
CREATE DIR "CHARLIE"  Enter
```

The directory structure is now as shown below:



Note that as long as the current mass storage volume is ":DOS,C" we did not need to include the MSVS, but only the path, in the CREATE DIR statement. In fact, the path may be omitted if you are creating files or directories in ("under") the current directory.

Creating Files and Other Directories Under a Directory

To create files subordinate to a new directory, you may either establish the new directory as the working directory or specify the directory path to that directory. Assuming your current working directory is "PROJECT.ONE", you could type:

```
MSI "CHARLIE" [Enter]
```

to move into the new directory, "CHARLIE".

You could verify the new working directory with a catalog listing by typing:

```
CAT [Enter]
```

The resulting listing would look like this:

```
DIRECTORY: C:\PROJECTS\PROJECT.ONE\CHARLIE
LABEL: HARD_DISK_C
FORMAT: DOS
AVAILABLE SPACE:     66896
                FILE     NUM   REC     MODIFIED
FILE NAME       TYPE    RECS   LEN     DATE     TIME PERMISSION
============ ====== ======= ===== ========= ===== ==========
```

As you can see from the listing, the directory contains no files or subdirectories. Let's create some files within "CHARLIE". To create an ASCII file named "ASCII_1" that is initially to contain 100 records, you would type:

```
CREATE ASCII "ASCII_1",100 [Enter]
```

To create a BDAT file named "BDAT_1" that is initially to contain 5 records, you would type:

```
CREATE BDAT "BDAT_1",5 [Enter]
```

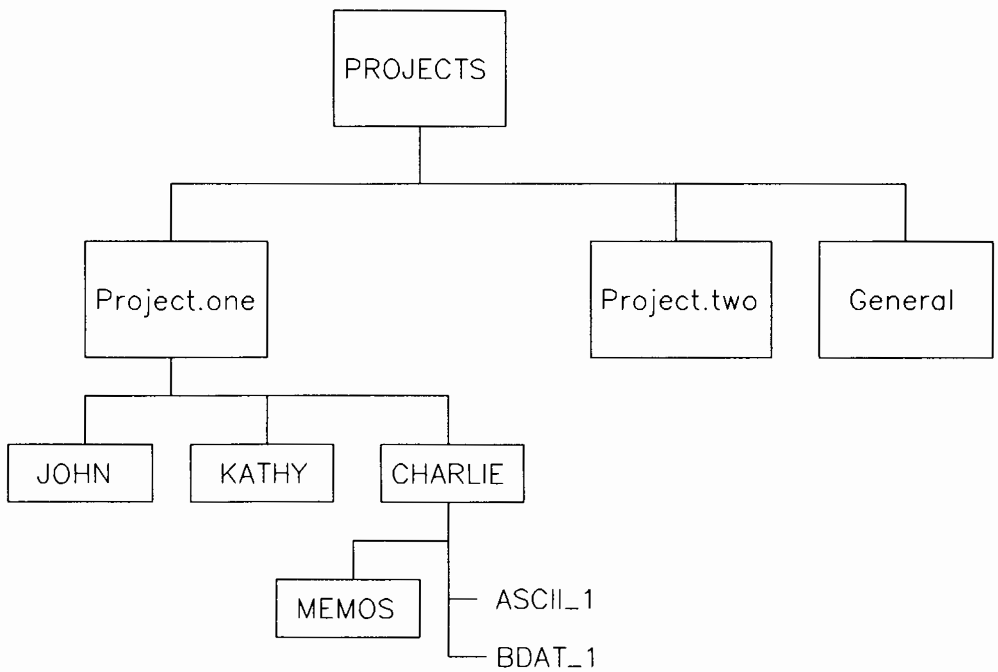
Note

There are four main file types in HP BASIC: PROG, BDAT, ASCII, and *untyped* DOS or HP-UX files. These file types are described in detail in *Programming with HP BASIC* (chapter 6, "Data Storage and Retrieval").

To create a subdirectory called "MEMOS" within "CHARLIE", you would type:

```
CREATE DIR "MEMOS"  [Enter]
```

The additions would make the directory structure look like this:



Now let's look at a CAT listing of "CHARLIE":

```
CAT [Enter]
```

```
DIRECTORY: C:\PROJECTS\PROJECT.ONE\CHARLIE
LABEL: HARD_DISK_C
FORMAT: DOS
AVAILABLE SPACE:     66776
               FILE     NUM   REC     MODIFIED
FILE NAME      TYPE     RECS  LEN     DATE     TIME  PERMISSION
============ ====== ======= ===== ========= ===== ==========
ASCII_1      ASCII     100   256 15-Apr-89 18:06  RW-RW-RW-
BDAT_1       BDAT        5   256 15-Apr-89 18:10  RW-RW-RW-
MEMOS        DIR         0     1 15-Apr-89 14:29  RWXRWXRWX
```

Protecting DFS Files

The HP BASIC "PROTECT" and "PERMIT" statements are not implemented for the DFS binary. *However, you can protect a DFS file from being purged or overwritten by setting the MS-DOS READ-ONLY attribute flag.* To do this, you can use the MS-DOS "ATTRIB" command. For example, to set the READ-ONLY attribute for the file "ASCII_1" in the directory "CHARLIE", exit BASIC and execute the following commands from the *MS-DOS command line*:

```
CD C:\PROJECTS\PROJECT.ONE\CHARLIE   [Enter]
ATTRIB +R ASCII_1   [Enter]
```

or:

```
ATTRIB +R C:\PROJECTS\PROJECT.ONE\CHARLIE\ASCII_1   [Enter]
```

In either case, a CAT of the directory "CHARLIE" will now show "`R--R--R--`" in the "PERMISSION" field for "ASCII_1":

```
DIRECTORY: C:\PROJECTS\PROJECT.ONE\CHARLIE
LABEL: HARD_DISK_C
FORMAT: DOS
AVAILABLE SPACE:     66776
                 FILE    NUM    REC      MODIFIED
FILE NAME        TYPE    RECS   LEN    DATE      TIME  PERMISSION
============ ====== ======= ===== ========= ===== ==========
ASCII_1          ASCII    100   256 15-Apr-89 18:06  R--R--R--
BDAT_1           BDAT       5   256 15-Apr-89 18:10  RW-RW-RW-
MEMOS            DIR        0     1 15-Apr-89 14:29  RWXRWXRWX
```

The file "ASCII_1" is now protected against being written to or purged. To remove this protection, execute:

```
ATTRIB -R C:\PROJECTS\PROJECT.ONE\CHARLIE\ASCII_1  Enter
```

from the MS-DOS command line. The "-R" option removes the READ-ONLY attribute. For further information about the ATTRIB command, refer to your MS-DOS *User's Reference* manual.

You don't have to exit HP BASIC to set the attribute flag. Instead, you can execute the MS-DOS "ATTRIB" statement using an "OUTPUT 19" statement from HP BASIC. To set the READ-ONLY attribute, execute:

```
OUTPUT 19; "ATTRIB +R C:\PROJECTS\PROJECT.ONE\CHARLIE\ASCII_1"  Enter
```

To remove the READ-ONLY attribute, execute:

```
OUTPUT 19; "ATTRIB -R C:\PROJECTS\PROJECT.ONE\CHARLIE\ASCII_1"  Enter
```

Copying DFS Files

The HP BASIC "COPY" statement allows you to copy a file to a new file name in the same directory, or to a different directory. (If you copy a file that has been write protected by setting its READ-ONLY attribute, the copy — the new file — will also be write protected.) For example, assuming that the current directory is still "CHARLIE", the HP BASIC statement:

```
COPY "BDAT_1" TO "\PROJECTS\PROJECT.ONE\KATHY\BDAT_1"  Enter
```

will copy the BDAT file named "BDAT_1" in "CHARLIE" to the directory "KATHY". The file name will be "BDAT_1" in both directories. (Files in different directories can have the same file name.) On the other hand, the statement:

```
COPY "BDAT_1" TO "BDAT_1.CPY"  Enter
```

Creates a copy of "BDAT_1" named "BDAT_1.CPY" in the current directory, "CHARLIE".

The statement:

```
COPY "DATAFILE:DOS,A" TO "\PROJECTS\PROJECT.ONE\CHARLIE\DATAFILE:DOS,C"  Enter
```

copies the file named "DATAFILE" from flexible disk "A:" to the directory "CHARLIE" on hard disk volume "C:". The following statement does the same as long as "CHARLIE" is the current directory.

```
COPY "DATAFILE:DOS,A" TO "DATAFILE"  Enter
```

For both the source and the destination file names in the HP BASIC "COPY" statement, if no path or volume is specified, the current mass storage directory is assumed.

Note The HP BASIC "COPY" statement can only be used to copy files *one-at-a-time*. You cannot copy an entire volume at once. For example, *the following statement will not work.*

```
COPY ":DOS,A" TO ":DOS,B"
```

You can copy program files (the PROG file type) using the COPY statement. However, you can also use the LOAD and STORE statements to copy programs by loading them into memory and then storing them to another mass storage location. For example, if the program file "TEST_PRG" is in the directory "C:\BLP\PROGS", the following HP BASIC statements will make a copy on a flexible disk in drive A:

```
LOAD "\BLP\PROGS\TEST_PRG:DOS,C"  Enter

STORE "TEST_PRG:DOS,A"  Enter
```

Purging DFS Files and Directories

You can use the HP BASIC "PURGE" statement to remove any file that is not write protected (refer to "Protecting DFS Files"). For example, to remove the file "BDAT_1" from the directory "CHARLIE", type:

```
PURGE "\PROJECTS\PROJECT.ONE\CHARLIE\BDAT_1:DOS,C"   Enter
```

Or you could make "CHARLIE" the current directory and purge as follows:

```
MSI "\PROJECTS\PROJECT.ONE\CHARLIE:DOS,C"   Enter

PURGE "BDAT_1"  Enter
```

You can use the PURGE statement to remove a directory from mass storage, provided you remove all files and subdirectories from it first. For example, to purge the "MEMOS" directory, type:

```
PURGE "MEMOS"   Enter
```

provided "CHARLIE" is the current directory, or else type:

```
PURGE "\PROJECTS\PROJECT.ONE\CHARLIE\MEMOS:DOS,C"   Enter
```

Once you have purged all of the files and subdirectories from "CHARLIE", you can purge "CHARLIE" itself. However, you cannot do this if "CHARLIE" is still the current directory. To return to the directory immediately above "CHARLIE" ("PROJECT.ONE"), type:

```
MSI ".."   Enter
```

Now you can remove the directory "CHARLIE":

```
PURGE "CHARLIE"   Enter
```

Additional I/O Path Registers for DFS

For information about I/O path registers for mass storage files, refer to the "Interface Registers" appendix in the HP BASIC *Language Reference* manual. In general the DFS binary uses Status and Control Registers 0 through 8 exactly as described in that manual. However, Status Register 2 returns the MS-DOS drive number (0 = A:, 1 = B:, 2 = C:, and 3 = D:), rather than a select code. Also, note that the register descriptions given for the "HP-UX" file type are applicable to the "DOS" file type for DFS. The DFS binary defines some additional I/O path registers, which are described here.

Control and Status Registers 9 and 10 are used by the DFS binary to control output to DOS, BDAT, and ASCII data files. In the default mode, when you output data to a file that has been assigned to an I/O path name, the data is not written to the disk file immediately. Instead, the data is held in a buffer until the buffer is full. When the file is closed, all data still in the buffer will be written to the disk file. This process increases throughput, but there is a potential for data loss (if there is a power interruption, for example). I/O Path Registers 9 and 10 allow you to select the mode of operation.

Control Register 9

The default value of this register is 1, which specifies that I/O path output is buffered. If you write a 0 to this register, the unbuffered mode is selected. All data will be written directly to the disk file. If you write a 1 (or any non-zero value) to this register, buffered mode is restored.

Status Register 9

A value of 1 indicates I/O path output is buffered. A value of 0 indicates that it is not buffered.

Control Register 10

Writing a 1 to this register causes the pending buffer to be written to the disk file and the directory entry for the file to be updated. However, this register has no effect on the buffering mode as defined by Control Register 9.

You can use the buffered mode to obtain high throughput, but occasionally write a 1 to Control Register 10 to synchronize data in the buffer and file.

Status Register 10

The value of this register is not defined.

Note

You should specify the "unbuffered" mode if an HP BASIC program is running in background and outputting data to a DOS file, and a foreground program is reading the file. Otherwise, the DOS file will not have all of the latest data available when it is read. (Alternatively, you could use the buffered mode and occasionally write a 1 to Control Register 10 to synchronize the buffer and file.)

If you redirect CRT output to a DFS file with the PRINTER IS statement, the *buffered* mode of output is used. To synchronize the disk file, you must execute a PRINTER IS statement to another device (for example, the CRT).

Using the LIF and HPW File Systems

For most applications you will want to use the DFS binary as your primary mass storage file system. However, to provide compatibility with other systems, the measurement coprocessor supports two additional file systems:

- *The HP 9000 Series 200/300 LIF file system.* The measurement coprocessor supports the LIF file system for compatibility with HP 9000 Series 200 and Series 300 computers.
- *The HPW (virtual-LIF) file system.* The measurement coprocessor supports the HPW file system for backward compatibility with earlier versions of the measurement coprocessor (language processor) software.

LIF and HPW File Names

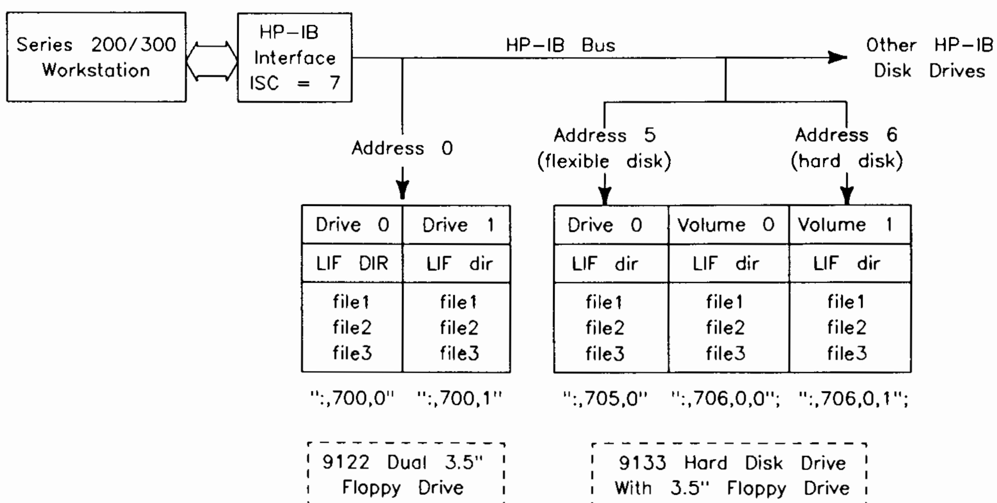
In the LIF and HPW file systems, file names may have up to 10 characters. However, no extensions are allowed. LIF file names are "case sensitive." That is, both uppercase and lowercase characters may be used to make different file names.

The HP 9000 Series 200/300 LIF File System

The LIF (Logical Interchange Format) file system is supported by HP 9000 Series 200/300 computers, the HP Model R/322 Controller, and all versions of the measurement coprocessor. In fact, since all HP BASIC systems can access LIF-formatted flexible disks, they are useful for transferring files to other computer systems such as the HP Model R/332 Controller. Note, however, that LIF formatting is completely different from MS-DOS formatting. *You cannot access a LIF disk from MS-DOS, only from HP BASIC.*

Accessing External LIF Disk Drives

The Series 200/300 mass storage system is based on external HP-IB disk drives. The HP-IB drives can be flexible disk drives, hard disk drives, or combination flexible and hard disk drives. Each drive is designated by a separate address on the HP-IB bus. A combination drive may have two addresses; one for the flexible, another for the hard disk. A typical configuration is shown below.



In the LIF file system, unlike the DOS file system, each volume stands alone — there is no hierarchy of directories. Each volume contains a collection of files formatted in the Hewlett-Packard LIF format. Also, each volume has a **LIF directory** file that identifies the other files in the volume. It is this LIF directory that contains the address of each file, the file type, number of records, and so forth.

You can connect external flexible and hard disk drives to the HP-IB connector on the back panel of your measurement coprocessor. You can also connect external drives to a separate HP-IB card in another PC slot.

The measurement coprocessor accesses external HP-IB disk drives exactly as an HP 9000 Series 200/300 computer would. Each disk volume is accessed by referencing its Mass Storage Volume Specifier, or MSVS. (The MSVS is often called the Mass Storage Unit Specifier, or MSUS.) The MSVS incorporates the drive type, HP-IB interface select code, and the address of the drive unit. In the MSVS:

```
":CS80,700,0"
```

the drive type is "CS80", the HP-IB interface select code is "7", the primary address of the drive is "00", and the drive unit number is "0".

The default drive type is "CS80", so you can usually leave out the drive type. Thus, for an HP 9122 Dual 3.5-Inch Flexible Disk Drive at select code "7", primary address "00":

`":,700,0"` is the MSVS for "drive unit 0" (the left drive), and

`":,700,1"` is the MSVS for "drive unit 1" (the right drive).

You can select an external LIF drive by using the correct MSVS in a MASS STORAGE IS statement. Referring to the figure on the previous page, use the HP BASIC command:

```
MASS STORAGE IS ":,705,0"  Enter
```

to select the flexible disk drive on the HP 9133 combination drive. Use the command:

```
MASS STORAGE IS ":,706,0,1"  Enter
```

to select the second volume of the hard disk drive.

For further information on the syntax of MASS STORAGE IS, refer to the HP BASIC *Language Reference* manual.

Files on drives other than the current mass storage volume can be accessed by referencing the appropriate MSVS. For example, to load the program "Test_1" from the flexible disk volume at ":,700,0", use the HP BASIC command:

```
LOAD "Test_1:,700,0"  (Enter)
```

Note

In the LIF file system, file names may have up to 10 characters. However, no extensions are allowed. LIF file names are "case sensitive." That is, both uppercase and lowercase characters may be used to make different file names. Refer to the *Porting Guide for the Measurement Coprocessor* for additional information about file name compatibility.

The measurement coprocessor can be configured to directly control up to two HP-IB interface cards in PC slots. Select codes 24 and 25 are reserved by the measurement coprocessor for HP-IB cards. However, to access these cards you must first configure them with the configuration utility "CONF.EXE" (refer to chapter 8). Once you have configured an HP-IB card at select code 24 or 25, you can access disk drives connected to that card in the usual manner. For example:

```
MSI ":,2407,1"  (Enter)
```

selects drive unit "1" on an HP-IB disk drive at select code "24", primary address "07".

Accessing LIF Disks in the Internal PC Disk Drives

You can access LIF-formatted flexible disks in the internal disk drives of your PC. The measurement coprocessor accesses LIF volumes in the internal PC disk drives as though they were HP-IB disk drives at interface select code 15. However, each drive or directory to be accessed through select code 15 must first be configured using the HP BASIC configuration utility. Chapter 8, "Changing the HP BASIC Configuration," tells how to use the configuration utility to configure one or more drives. The MSVS ":,1500,0" is used for the first drive in the configuration list, ":,1500,1" is used for the second, and so forth, on up to ":,1500,14".

Note You don't need to configure the internal disk drives for DFS file access, only for LIF and HPW file access.

For example, if you configure disk drive "A:" as the first drive in the list, the following command makes a LIF disk in drive "A:" the current mass storage:

```
MSI ":,1500,0" [Enter]
```

If drive "B:" is the second drive in the configuration list, it can be accessed with the command:

```
MSI ":,1500,1" [Enter]
```

You can also specify the MSVS in any statement that accesses a file. For example, to LOAD the program "Test_1" from a LIF disk at ":,1500,0", use the command:

```
LOAD "Test_1:,1500,0"  [Enter]
```

To obtain a CAT listing of ":,1500,0" execute:

```
CAT ":,1500,0"  [Enter]
```

or:

```
MSI ":,1500,0"  [Enter]
CAT  [Enter]
```

The following is a typical listing:

```
:CS80, 1500, 0
VOLUME LABEL: B9826
FILE NAME PRO TYPE  REC/FILE BYTE/REC   ADDRESS     DATE     TIME

Test_1        PROG        14      256        16    5-Apr-89 19:00
Datafile      ASCII        4      256        30    5-Apr-89 19:01
DATAFILE      BDAT         5      256        34    5-Apr-89 19:03
DATAFILE03    HP-UX       94        1        39    5-Apr-89 19:05
```

Initializing a LIF Disk

Just as MS-DOS disks must be formatted, LIF disks must be *initialized* before use. The initialization process consists of formatting the disk and creating the LIF directory. If you are using an external HP-IB disk drive, the initialization procedure is the same as for a Series 200/300 system — you use the HP BASIC "INITIALIZE" statement. For example:

```
INITIALIZE ":,700,0",0  [Enter]
```

initializes a disk in a drive at select code "7", address "00", unit number "0" with the default interleave factor. The complete syntax of the INITIALIZE statement is covered in the HP BASIC *Language Reference* manual.

If you want to initialize a LIF disk in one of the internal PC disk drives, you will have to use the "LIFINIT" utility. This is because a statement such as `INITIALIZE ":,1500,0"` will create an HPW volume, *not* a LIF volume.

To use the “LIFINIT” utility, insert your “Manual Examples, LIFINIT, and Selected CSUBs” disk in drive A: and type:

`A:LIFINIT` (Enter)

The following screen will be displayed.

```
Hewlett-Packard PC LIFINIT Utility, Version A.00.02
Copyright (c) Hewlett-Packard Co. 1989. All rights reserved.


================================================================================
           DRIVE:   A:
       DISK TYPE:  5 1/4" flexible disk
     SECTOR SIZE:   256 bytes     (DISK CAPACITY =   264 KB)
      INTERLEAVE:  2
 DIRECTORY SIZE IN:
    NO. OF FILES:  112
    VOLUME LABEL:  HPBLP
================================================================================






Flexible disk drive in which disk will be initialized.


 NEXT    PREV                         START             DEFAULT
 CHOICE  CHOICE                       FORMAT            VALUE     EXIT
```

The LIFINIT program is “menu driven.” Just highlight the field that you want to change using the cursor-control (“arrow”) keys and use the `NEXT CHOICE` and `PREV CHOICE` softkeys ((F1) and (F2)) to select the parameter that you want. Use the `DEFAULT VALUE` ((F7)) softkey to return to the default value for a given field.

For example, if you want to initialize a 3.5-inch high density LIF disk in drive "B:", just highlight the "DRIVE" field and press NEXT CHOICE to select "B:".

Then highlight the "DISK TYPE" field and press NEXT CHOICE three times until the following screen appears.

```
Hewlett-Packard PC LIFINIT Utility, Version A.00.02
Copyright (c) Hewlett-Packard Co. 1989. All rights reserved.


==============================================================================
           DRIVE:   B:
       DISK TYPE:  3 1/2" high density (black disk)
     SECTOR SIZE:   256 bytes     (DISK CAPACITY =  1232 KB)
      INTERLEAVE:  2
 DIRECTORY SIZE IN:
     NO. OF FILES:  240
     VOLUME LABEL:  HPBLP
==============================================================================
```

Note that when you change the "DISK TYPE" parameter, the "DISK CAPACITY" and "NO. OF FILES" fields automatically change to reflect the capacity of the disk type selected.

If you want to change the "SECTOR SIZE," "INTERLEAVE," or "NO. OF FILES" parameter, highlight the appropriate field and use the softkeys to select the value you want. (Note that for 3.5-inch disks the 1024-byte sector size gives higher total disk capacity.) Generally, however, you will want to use the default parameters for your disk type. For further information on these parameters, refer to the INITIALIZE statement section in the HP BASIC *Language Reference* manual.

You can initialize the disk using the default volume label ("HPBLP"), or you can highlight the "VOLUME LABEL" field and type in a label of up to six characters. The volume label must start with an alpha character but may otherwise be any combination of alphanumeric characters. (No embedded blanks are allowed.)

Once you have selected the initialization parameters for your disk, insert the disk to be initialized in the drive you have chosen and press START FORMAT (F5). When the process is complete, press EXIT (F8) to terminate the program.

Note

Any LIF-formatted flexible disk may be used in either an external, HP-IB disk drive or an internal, PC disk drive (assuming the disk is of the correct size and type, of course). It does not matter whether the disk was initialized with LIFINIT, or in an external drive with the INITIALIZE statement.

The HPW (Virtual-LIF) File System

The measurement coprocessor provides the HPW (virtual-LIF) file system for backward compatibility with previous versions. If you have been using an earlier version (A.00 through A.02) of the software, you can access your HPW files from the current version (C.00 or higher). *However, we recommend that you use DFS as your primary file system.* The DFS binary, described earlier in this chapter, provides higher mass storage performance and allows direct MS-DOS access of HP BASIC data files.

The HPW file system is called a "virtual-LIF" file system because it has LIF characteristics, but exists in the MS-DOS environment. An HPW volume exists on an MS-DOS formatted disk, but HPW files have a different internal organization than regular MS-DOS files. Each HPW volume has a virtual-LIF directory file named "HPWLIF.DIR". This HPWLIF.DIR file, like a LIF directory, contains the address and file type of each HPW file.

You can only create an HPW volume on an MS-DOS formatted disk medium. Thus, HPW volumes can be created only in the PC's MS-DOS disk drives, not in external LIF disk drives.

Accessing HPW Disk Volumes

HPW disk volumes can only be accessed in the internal PC disk drives through select code 15. Although HPW volumes share characteristics of LIF and MS-DOS volumes, you can access them from HP BASIC exactly as you would access a LIF volume in an internal PC disk drive.

The measurement coprocessor accesses HPW volumes in the internal PC drives as though they were HP-IB disk drives at interface select code 15. However, each drive or directory to be accessed through select code 15 must first be configured using the HP BASIC configuration utility. Chapter 8, "Changing the HP BASIC Configuration," tells how to use the configuration utility to configure one or more drives or directories. The MSVS ":,1500,0" is used for the first drive in the configuration list, ":,1500,1" is used for the second, and so forth, on up to ":,1500,14".

Note

You don't need to configure the internal disk drives for DFS file access, only for LIF and HPW file access.

For example, if you configure disk drive "A:" as the first drive in the list, its MSVS is ":,1500,0". You can LOAD the program "Test_1" in either of the following ways:

```
MSI ":,1500,0"  [Enter]
LOAD "Test_1"  [Enter]
```

or:

```
LOAD "Test_1:,1500,0"  [Enter]
```

If you configure the directory "C:\HPW" as the second drive in the list, it can be accessed as follows:

```
MSI ":,1500,1"  [Enter]
```

To obtain a CAT listing of ":,1500,0" execute:

```
CAT ":,1500,0"  [Enter]
```

or:

```
MSI ":,1500,0"  [Enter]
CAT  [Enter]
```

The following is a typical listing:

```
:CS80, 1500, 0
VOLUME LABEL: B9826
FILE NAME PRO TYPE  REC/FILE BYTE/REC   ADDRESS    DATE     TIME

Test_1        PROG        14      256        16   5-Apr-89 19:20
Datafile      ASCII        4      256        30   5-Apr-89 19:22
DATAFILE      BDAT         5      256        34   5-Apr-89 19:23
DATAFILE03    HP-UX       94        1        39   5-Apr-89 19:25
```

Note

In the HPW (virtual-LIF) file system, file names may have up to 10 characters. However, no extensions are allowed. HPW file names, like LIF file names, are "case sensitive." That is, both uppercase and lowercase characters may be used to make different file names. Refer to the *Porting Guide for the Measurement Coprocessor* for additional information about file name compatibility.

Initializing an HPW Disk

You can initialize an HPW disk volume with the INITIALIZE statement. To initialize a flexible disk in drive "A:" (provided it is configured as MSVS ":,1500,0"), execute the following HP BASIC statement:

```
INITIALIZE ":,1500,0"
```
(Enter)

The INITIALIZE command will cause the following to occur:

- The flexible disk will be formatted as an MS-DOS disk.
- The virtual-LIF directory "HPWLIF.DIR" will be created.

Note

You can use any MS-DOS formatted disk as an HPW disk. The first time you access an MS-DOS formatted disk volume or directory from HP BASIC through select code 15, "HPWLIF.DIR" will be created if it doesn't already exist in the volume or directory. Thus, for example, you could format a disk in drive "A:" with the MS-DOS command "FORMAT A:" and then use that disk as an HPW volume.

Precautions Regarding HPW Files

As described earlier, the HPW file system is actually a special "virtual-LIF" directory system that exists on an MS-DOS formatted disk. This has some implications that you should consider.

1. HP BASIC cannot recognize HPW files in a virtual-LIF volume unless they have an entry in the volume's HPWLIF.DIR file. Thus, you should not use the *MS-DOS* "COPY" command to copy an HPW file into a virtual-LIF volume. HP BASIC won't recognize the copied file since HPWLIF.DIR won't be updated. HPW files should be copied from virtual-LIF volume to virtual-LIF volume by using the *HP BASIC* "COPY" command.
2. Similarly, an HPW file should not be deleted using the MS-DOS "DEL" command since that does not remove the entry from the HPWLIF.DIR file. In fact, all HPW files are write-protected from MS-DOS to prevent this. However, it is possible for this write-protection to be removed. Use the HP BASIC "PURGE" command to remove files.
3. Since virtual-LIF drives are simply MS-DOS directories containing HPW files, any directory in the MS-DOS file system — whether it's on a flexible disk, a hard disk, a PC RAM disk, or a PC network mass storage server — can be used as a virtual-LIF drive. If there is no HPWLIF.DIR file on a disk or directory designated as a virtual-LIF drive, an HPWLIF.DIR file will automatically be created when HP BASIC attempts to access it. Note that in this case an error will result if the disk is write-protected.
4. HPW file names, when accessed from HP BASIC, are just like LIF file names. They consist of up to ten alphanumeric characters and are case sensitive. However, when you list the MS-DOS directory of a virtual-LIF volume, the file names are mapped according to the following conventions;
   - Embedded blanks and illegal MS-DOS characters become "_".
   - Lowercase letters are converted to uppercase letters.
   - Filenames greater than eight characters appear with MS-DOS extensions. For example, "ABCDEFGHIJ" would be renamed in MS-DOS as "ABCDEFGH.IJ".
   - To avoid duplicate names in MS-DOS, a unique two-letter extension is appended to the MS-DOS file name when necessary.

5. *Avoid accessing a virtual-LIF volume through the DFS binary.* The DFS binary directly accesses the MS-DOS file system from HP BASIC. Thus, if you have a virtual-LIF volume in the directory "C:\HPW", you could access that volume by executing:

   ```
   MSI "\HPW:DOS,C"  Enter
   ```

   If you do this, there are several possible problems:

   - A CAT of the directory from DFS will indicate that all HPW files have the DOS file type. This is because the file type information is in the HPWLIF.DIR file, and is unavailable to DFS.
   - As mentioned previously, the internal structure of HPW files is different from that of an MS-DOS file, and DFS expects an MS-DOS internal organization.
   - In many cases the file protection against MS-DOS access to HPW files may not protect against DFS access. Thus, you may accidentally purge an HPW file. *If you purge the HPWLIF.DIR file, you will lose access to all of the HPW files in the virtual-LIF volume.*
   - DFS follows the file naming conventions of MS-DOS, which has the implications noted previously.

Because of the above considerations, the best procedure is to use DFS as your primary file system. You may access your existing HPW files in virtual-LIF volumes by configuring the appropriate drives, as described previously. *Although you could continue to access the HPW files on a day-to-day basis, it is recommended that you simply copy all of your HPW files into a DFS directory.* Once you have done this you won't have any further problems with file compatibility. The copying process automatically translates the files to the MS-DOS format for DFS.

You can use the HP BASIC "COPY" statement to copy HPW files one at a time to a DFS directory (refer to "Copying Files Between File Systems"). However, you may find it more convenient to use the HP BASIC utility program called "CATCOPY" to batch copy all of your HPW (or LIF) files. This program is described in the *Porting Guide for the Measurement Coprocessor*.

Deleting an HPW Volume

As previously mentioned, HPW files are write-protected against accidental access or deletion from MS-DOS. (The MS-DOS READ-ONLY attribute is set in all HPW files.) However, in some cases, a user may want to clear the contents of an MS-DOS disk or directory designated as a virtual-LIF volume, and in that case the write-protection becomes a nuisance. Fortunately, write-protection can be removed using the MS-DOS "ATTRIB" command. This command is described in detail in your MS-DOS *User's Reference* manual.

Caution The procedure below will delete all files from the specified directory. Make sure that you have copied all of your data and program files from the directory, or that you don't need them any longer, before doing this.

As an example, suppose that you have copied all of your data and program files from the directory "C:\HPW" to a DFS directory, and that you want to delete all of the files from "C:\HPW". You can remove write protection from all files in "C:\HPW" with the MS-DOS command:

```
ATTRIB -R C:\HPW\*.* (Enter)
```

To delete all of the files, execute an MS-DOS "DEL" command:

```
DEL C:\HPW\*.* (Enter)
```

To remove the directory, execute:

```
RMDIR C:\HPW (Enter)
```

from MS-DOS.

Protecting LIF and HPW Files

You can protect files in the LIF and HPW file systems by using the HP BASIC "PROTECT" statement to establish protect codes. (The PROTECT statement can also be used in the SRM environment. However, the PROTECT statement format for remote files is different. Refer to appendix A, "Using SRM with the Measurement Coprocessor," for further information.)

Note

Although you can use PROTECT for the HPW file system, this form of file protection only prevents access from within HP BASIC. It does not prevent read/write access of HPW files from MS-DOS, outside of BASIC. Also, the PROTECT statement cannot be used to protect files in the DFS file system.

Protect codes are two-character strings that can be assigned to any BDAT, BIN or PROG type file with the PROTECT statement. Protect codes are not unbreakable; they are only intended to prevent accidentally writing in files and directories.

For instance, the following statement assigns the protect code "AA" to the file named "FILE1".

```
PROTECT "FILE1","AA"  Enter
```

File specifiers in mass storage statements that write to a file or directory must include the protect code if the file has one. Mass storage statements that read a file or directory (CAT, LOAD, LOAD BIN, LOADSUB ALL FROM, GET and COPY) do not require the protect code. A protect code is specified by placing it in brackets right after the file name. To assign an I/O path name to the file named "FILE1," you would now have to include the protect code.

```
ASSIGN @Path1 TO "FILE1<AA>"  Enter
```

If you assign a protect code longer than two characters, the system will ignore everything after the second (non-bland) character. For example, the protect codes LONGPASS, LOLLYPOP, and LOST all result in the same protect code: "LO". This rule holds both for protecting a file and for specifying the protect code in a file specifier. For example:

```
PROTECT "FILE1","Protect1"  Enter
```

assigns the protect code "Pr" to "FILE1".

To rename the file, you could use the following HP BASIC command:

```
RENAME "FILE1<Prattle>" TO "FILE2"  Enter
```

"Prattle" is an acceptable protect code, since it starts with "Pr." Note that we do not include a protect code in the new file name. If you do, the system ignores it since the old protect code is passed to the new file name. FILE2 still has the protect code "Pr". To rename the file again, we might write:

```
RENAME "FILE2<Pr>" TO "FILE3"  Enter
```

Renaming a file has the effect of changing the file name in the directory and leaving everything else intact.

In addition to using the PROTECT statement, you can also assign a protect code to a BDAT file when you create it. For example:

```
CREATE BDAT "Example<xx>",10  Enter
```

creates a 10-record BDAT file called "Example" and gives it a protect code of "xx". You can also do this to PROG files with the STORE and STORE BIN statements. However, since ASCII files cannot be protected, a protect code cannot be included in any CREATE ASCII, SAVE, or RE-SAVE statement.

To change a protect code, simply execute a new PROTECT statement. To change the protect code of "Example" to "yy," execute:

```
PROTECT "Example<xx>","yy"  Enter
```

Note that you must include the current protect code in the file specifier.

To completely remove a protect code from a file, protect the file with a code consisting of two blanks. For example, to remove the protect code from file "Example," execute:

```
PROTECT "Example<yy>","  "  Enter
```

When specifying a file that does not have a protect code, you can either ignore the code entirely or include a code of two spaces. You can use either of the following commands to PURGE the file "Example":

```
PURGE "Example"  [Enter]

PURGE "Example<  >"  [Enter]
```

Copying Files Between File Systems

The HP BASIC "COPY" statement allows you to duplicate individual files. Any type of file may be copied. You can use COPY within a volume or directory, or to copy files from one volume or directory to another. *In fact, the "source" and "destination" volumes may be in different file systems.* Thus, you can use the COPY statement to perform the conversion of HPW or LIF files to the DFS file system.

Note

You can use the HP BASIC "COPY" statement to copy *individual* files only. You *cannot* use "COPY" to copy an entire volume or directory. If you want to copy all files from an HPW or LIF volume to a DFS directory, use the "CATCOPY" utility provided with HP BASIC. This utility and its use is described in the *Porting Guide for the Measurement Coprocessor*.

Let's look at some examples:

The following command copies "File1" from the current system mass storage device to a new file called "File2" on the same mass storage.

```
COPY "File1" TO "File2"  [Enter]
```

This form of the COPY statement doesn't help much if you want to copy files from one file system to another. However, you can add a volume and/or path name to both the source and destination file names. For example, to copy "File1" from a LIF volume (":,1500,0") to a DFS directory (C:\BLP), you could use the command:

```
COPY "File1:,1500,0" TO "\BLP\FILE1:DOS,C"  [Enter]
```

Remember that DFS file names are case folded into uppercase, and that they have a maximum of eight characters plus a three character extension. If you execute the following statement:

```
COPY "Filename01:,1500,0" TO "\BLP\Filename01:DOS,C"  [Enter]
```

the destination file name will be truncated and case folded to "FILENAME". *If a file with the name "FILENAME" already exists, you will get a duplicate file name error.* You can avoid this problem by specifying a legal DFS file name for the destination file:

```
COPY "Filename01:,1500,0" TO "\BLP\FILENAME.01:DOS,C"  [Enter]
```

You may find it more convenient to make the destination directory the current mass storage, and then copy one or more files. For example:

```
MSI "\BLP:DOS,C"  [Enter]
COPY "Filename01:,1500,0" TO "FILENAME.01"  [Enter]
COPY "Filename02:,1500,0" TO "FILENAME.02"  [Enter]
```

Note The procedure for copying an HPW file to a DFS directory is the same as for a LIF file. Note, however, that the HPW or LIF volume must be configured before you can access it.

You can also use COPY to transfer files from a DFS directory to a LIF or HPW disk. For example, if you want to copy the file "TEST_1" from the current DFS directory to a LIF initialized disk in drive "A:" (configured as ":,1500,0"), you could use the command:

```
COPY "TEST_1" TO "TEST_1:,1500,0"  [Enter]
```

The file system translation occurs automatically, regardless of the direction of transfer. The COPY command reads the source file using the protocols of the source volume or directory, and writes the destination file using the protocols of the destination volume or directory. In fact, the above statement will also work for an HPW destination disk in drive "A:" since COPY will recognize the format of the disk at MSVS ":,1500,0".

You can also use the HP BASIC "LOAD" and "STORE" commands to transfer program files. For example, if the current mass storage volume is a DFS directory containing the file "TEST_1", you could execute the following HP BASIC statements:

```
LOAD "TEST_1"  (Enter)
STORE "TEST_1:,1500,0"  (Enter)
```

The result is the same as for the previous "COPY" command.

Using RAM Volumes

Although mass storage is traditionally implemented using magnetic media (disk, drum, or tape), the file management protocols can be applied to any device which stores data, including computer memory, or RAM. Areas of computer memory may be treated as mass storage devices. These **RAM volumes** are volatile (all information is lost when the power goes off), but provide higher speed than magnetic media.

A typical RAM-volume application is to copy a disk volume into memory, perform all necessary manipulations using the RAM volume, then copy the new information back to disk.

Although all mass storage operations work with RAM volumes, only certain applications benefit from this technique.

You can create HP BASIC RAM volumes using the INITIALIZE statement. A special form of this statement is used, with a unit size parameter in the position normally occupied by the interleave factor. The device type is always "MEMORY", and the device selector is always 0. Unit numbers 0 thru 31 may be used. For example, the HP BASIC statement:

```
INITIALIZE ":MEMORY,0,7" ,220  (Enter)
```

creates a RAM volume that is 220 sectors long with unit number 7. Note that the unit size parameter is in 256-byte sectors, just as for a LIF volume.

If the unit size parameter is omitted, the result is a RAM volume that is the same size as a 5.25-inch disk: 1056 sectors, or 270,336 bytes. The default size RAM volume provides only 80 directory entries, while the disks may contain up to 112 directory entries. If a disk is copied into the RAM volume, the entire directory will be copied.

The unit size of a RAM volume must be at least four sectors and can be as large as available memory permits. Two sectors are taken for system use, and about one sector of directory is created for each 100 sectors of unit size.

No RAM volumes exist at power-up or after a SCRATCH A. It is recommended that all binaries be loaded before RAM volumes are initialized. If a binary is loaded after a RAM volume is initialized, the memory used for the RAM volume cannot be recovered until the computer is turned off and back on again (or rebooted).

A RAM volume can be re-initialized to the same or a different size. If the size is different, memory space may be lost until the next SCRATCH A.

After they are created, RAM volumes are accessed by using their unit number in a MEMORY media specifier. The following examples show typical mass storage unit specifiers for a RAM volume with unit number 7.

```
MASS STORAGE IS ":MEMORY,0,7"  (Enter)
```

or

```
ASSIGN @Ram TO "TEMP:MEMORY,0,7"  (Enter)
```

Note

The above process creates a RAM disk in the *measurement coprocessor* RAM. This RAM disk *cannot* be directly accessed from MS-DOS. On the other hand, an MS-DOS RAM disk *can* be accessed from the measurement coprocessor just like any other MS-DOS mass storage device. You can set up an MS-DOS RAM disk using the VDISK or RAMDRIVE driver. Refer to your MS-DOS documentation for further information.

8

Changing the HP BASIC Configuration File

When you boot HP BASIC, a configuration file is used to tell the system what resources are available to your computer and how they are to be used. The file name "BLP.CON" is reserved for the configuration file. BLP.CON must be in the same directory as the boot program, or it must be specified in the MS-DOS search path.

The default configuration file, BLP.CON, is found on measurement coprocessor software disk one. This file is installed automatically into the HP BASIC system directory (normally C:\BLP) when you run the INSTALL program. This configuration file will work with most PC applications, however you may wish to create a custom configuration that matches your needs. For example, you may want to modify the configuration to access optional interfaces, to configure internal disk drives to access LIF or HPW files, or to modify key definitions. (On rare occasions an incorrect BLP.CON configuration may prevent you from successfully booting HP BASIC. If this happens, compare your hardware configuration with the HP BASIC configuration and correct any discrepancies.)

You can modify the HP BASIC configuration (BLP.CON) file using the configuration utility "CONF.EXE". This utility is loaded into the HP BASIC system directory (normally C:\BLP) when you run the INSTALL program.

Note

If you have been using an earlier version of the software that used a configuration file named "HPW.CON", you can preserve its configuration for use with your new HP BASIC version. Refer to "Compatibility with Previous-Version Configuration Files" at the end of this chapter.

Running the Configuration Utility Program

To use the configuration utility program, CONF.EXE, you must exit BASIC to the MS-DOS command line. Change directories to your HP BASIC system directory (normally "C:\BLP"):

```
CD C:\BLP  (Enter)
```

and then start the configuration utility program:

```
CONF  (Enter)
```

The configuration program will read the current configuration from the BLP.CON file, and will allow you to change the configuration and store those changes back into BLP.CON. *You must invoke CONF.EXE from the BASIC system directory (C:\BLP) or the program won't be able to find the correct configuration file.*

Note

You can execute the configuration utility program with an optional parameter specifying a configuration file to use instead of BLP.CON. (If no file is specified, BLP.CON is automatically used.) To execute the configuration utility program using an optional configuration file, follow "CONF" with a path name (optional) and file name. For example, if you want to create a new configuration file named "NEW.CON", execute either:

```
CONF NEW.CON  (Enter)
```

or

```
CONF C:\BLP\NEW.CON  (Enter)
```

The configuration program will allow you to select a new configuration and store it in the file NEW.CON. However, *HP BASIC only recognizes the file named "BLP.CON" as the configuration file.* If you create a file with a different name, you will have to rename it "BLP.CON" in order to make the new configuration active. You can do this with the MS-DOS "RENAME" command. For example, type:

```
RENAME NEW.CON BLP.CON  (Enter)
```

Once the program is running, you will see a screen similar to the one below. Note that the screen is divided into four areas. On the left is a column labeled "Primary Configurations." On the upper right is the "Serial Cards" area. On the lower right is the "HP-IB Cards" area. Along the bottom are labeled function keys. The following screen shows the default configuration provided in the BLP.CON file loaded by the INSTALL program:

```
                 Measurement Coprocessor Configuration Utility   B.00.01

        Primary Configurations                          Serial Cards

DOS cmd save mode: ON                com port: COM1              NONE
 DOS command wait: ON                sel code: 9
Mouse sensitivity: X=01 Y=00        int level: 6
            cache: OFF                   baud: 9600
           drives: 0                   parity: DIS  ODD
          keydefs: 0                 char len: 8
                                    stop bits: 1
                             modem disconnect: CD RI DSR CTS

                                                  HP-IB Cards

                                  PC sel code: NONE              NONE
                                 PC int level:
                                     sel code:
                                    int level:

ON saves/restores BASIC alpha display during DOS command execution

  NEXT     PREV                          SAVE              DEFAULT
 CHOICE   CHOICE                        CONFIG              VALUE     EXIT
```

The configuration utility, CONF.EXE, is "menu driven." Just use the cursor-control keys (the "arrow" keys) to move the highlight from field to field throughout the screen. Use function keys F1 and F2 (NEXT CHOICE and PREV CHOICE) to progress through the different choices of values for the current field (the field where the cursor is currently located). If you want to return to the default value for the current field, just press F7 (DEFAULT VALUE). Once you have selected the configuration information for all fields, just press F5 (SAVE CONFIG) to save the information into the BLP.CON file. To return to MS-DOS, press F8 (EXIT).

Now let's look at the configuration information represented in each field.

Primary Configurations

Each field in the Primary Configurations area is explained below. As you proceed through the items on the screen, read each explanation to find if you need to change the value. Use the function keys to select the appropriate value for your system configuration.

DOS Cmd Save Mode/DOS Command Wait

When you are working in HP BASIC and need to access MS-DOS, you will use the command "OUTPUT 19". "DOS cmd save mode" and "DOS command wait" affect computer behavior when you use "OUTPUT 19."

"DOS cmd save mode" has two possible values, ON and OFF.

- ON will provide a display for MS-DOS command output, but will clear the HP BASIC graphics display. HP BASIC alpha information will be retained.
- OFF will allow MS-DOS to write into the HP BASIC display, resulting in combined output. However, since the HP BASIC graphics display is not cleared in this mode, both HP BASIC alpha and graphics displays are retained and will be intact *provided* MS-DOS does not produce any display output.

"DOS command wait" has two values, ON and OFF. ON causes the computer to pause upon completion of an MS-DOS command. When a key is pressed, the computer will return to HP BASIC. OFF causes the computer to continue processing immediately.

Refer to "The MS-DOS Communications Port" in chapter 4 for more information.

Mouse Sensitivity

You need not use "Mouse sensitivity" for an HP-HIL mouse — only for a serial mouse that is compatible with the Microsoft mouse interface standard (INT 33H). The values set the x and y ratios of mouse movement "ticks" to screen pixel movement. (If you are using a serial mouse, you must "unconfigure" the serial port that it uses. Refer to "Serial Cards" later in this chapter.)

Cache

"Cache" is a buffer that can be used to store the equivalent of one disk sector of data. Cache can be either ON or OFF, and the default is OFF. Setting cache to ON may improve the speed of writing to disk data files in HPW volumes. *Cache only affects HPW mass storage operations, not LIF or DFS operations.*

When cache is ON, the buffer will store data until it is full. The data is then written onto a sector of the disk, and the buffer is cleared for more data. This method of saving data can be significantly faster than the when cache is OFF. However, if there is a power interruption, or a disk is removed, or the computer is rebooted before closing the file, then all of the data in cache will be lost.

Drives

If you intend to access LIF or HPW files on the internal PC disk drives, you will need to configure each drive or directory to be accessed. The default BLP.CON file does not configure any drives or directories for LIF or HPW file access. *The reason is that no drive configuration is necessary to access DFS files or directories.* The DFS binary provides much faster file access than either the LIF or HPW file systems. Also, DFS provides direct access to the MS-DOS file system, allowing easy transfer of files to MS-DOS applications. *Thus, DFS should be used as the primary file system for HP BASIC.*

For most applications it is recommended that you only configure those drives or directories that you actually plan to use for LIF or HPW file access. *If you don't have any LIF or HPW files, don't configure any drives or directories.* There are two reasons for this suggestion:

1. *Configuring drives and directories for LIF and HPW file access may cause a slight increase in the time that it takes to start up HP BASIC.* This is because the drive configuration introduces some additional "overhead" for mass storage access.
2. *If drives are configured for LIF and HPW file access, it is possible to accidentally access a drive or directory using the wrong file system.* The problem is that both the DFS and HPW file systems use MS-DOS directories, but it is risky to mix DFS and HPW files in the same directory.

Note

If you have been using the HPW file system with an earlier version of the measurement coprocessor (the HP 82300A/B BASIC Language Processor), you may have HPW data files stored in a directory on your hard disk, most likely "C:\HPW". If this is the case it is recommended that you *not* access "C:\HPW" through DFS. Instead, configure "C:\HPW" for HPW access, as explained in this section, and then copy all of your HPW files over to a DFS directory (for example, "C:\BLP" or "C:\BLP\PROGS"). You can use the HP BASIC "COPY" command (refer to chapter 7), or you can use the "CATCOPY" utility (refer to the *Porting Guide for the Measurement Coprocessor*) to do this. Once you have copied all of your files to the new directory, "un-configure" "C:\HPW".

When you highlight the "drives" field on the configuration screen, function key (F1) is labeled EDIT DRIVES. Press EDIT DRIVES and a window will appear in the middle of the screen. You will type the path names into this window. An example of the "drives" configuration screen is shown below:

```
              Measurement Coprocessor Configuration Utility   B.00.01

     :,1500, 0 A:
     :,1500, 1 C:\HPW
     :,1500, 2 _
     :,1500, 3
     :,1500, 4
     :,1500, 5
     :,1500, 6
     :,1500, 7
     :,1500, 8
     :,1500, 9
     :,1500,10
     :,1500,11
     :,1500,12
     :,1500,13
     :,1500,14

     Specifies mapping of MS-DOS disk drives to BASIC disk volumes

                                   DELETE   INSERT          Return
                                    ENTRY    ENTRY          to  MAIN
```

The measurement coprocessor can have 15 different path names specified for LIF or HPW file access. The path names are written just as they would be in MS-DOS. First, a drive name is specified and then, if desired, a directory (and possibly a sub-directory). Network devices and other forms of remote or peripheral devices may be included in drive specifications.

Note

If you are using the HP 82313A Hierarchical File System (HFS), you *must* specify a partition in addition to the drive name. Legal values for partitions are 2, 3, and 4. The partition then becomes a part of the drive name. For example, C2: is the second partition of drive C. Note that the colon is required following the partition number. Also, directories cannot be used with partitions.

Only one path name can be placed on each line in the window. Up to 15 lines may be used. The first path listed will be called "drive 0" by the measurement coprocessor. The second path will be "drive 1," the third will be "drive 2," and so forth.

In the example above, the first path entry, "A:", is configured to ":,1500,0" and the second entry, "C:\HPW", is configured to ":,1500,1". Thus, you can access LIF or HPW files on drive "A:" with the HP BASIC statement:

```
MSI ":,1500,0" Enter
```

and you can access HPW files in the directory "C:\HPW" with the statement:

```
MSI ":,1500,1" Enter
```

Note

For each physical disk drive, the drive list may include a drive specifier (C:) *or* up to 15 directories for that drive (C:\, C:\HPW, etc.), *but not both*. If a drive specifier (C:) is used, HP BASIC will access the directory that was the current MS-DOS directory for that drive when HP BASIC was started.

It is recommended that you not configure "C:\BLP" or any other DFS directory on your hard disk. Also, don't configure the whole "C:" hard disk volume. If you do, and you access a DFS directory through select code 15, an HPWLIF.DIR file will be created in the DFS directory. Mixing HPW and DFS files in the same directory is not recommended.

To type path names, use the keyboard typewriter keys. Either lowercase or uppercase letters may be used ("c:\hpw" is equivalent to "C:\HPW"). Use the (Ins) key to toggle between Insert and Replace modes:

- In Insert mode, characters are inserted before the character the cursor is currently on. The cursor is a block cursor, and the backspace key deletes characters.
- In Replace mode, the current character is replaced by the new character. The cursor is an underscore, and the backspace key acts the same as a left cursor control key.

The (DEL) key deletes the current character.

The "Drives" configuration screen provides three softkeys. The **DELETE ENTRY** softkey ((F5)) deletes the highlighted path name entry and "scrolls up" the remaining entries below it. The **INSERT ENTRY** softkey ((F6)) inserts a "blank" entry at the cursor and "scrolls down" the entries below it. When you finish typing path names into the window, press **Return to MAIN** ((F8)) to return to the main configuration screen again. Notice that there is a number next to the "drives" item. This number corresponds to the number of paths you have entered.

For example, suppose you want to add drive "B:" to the above example configuration. Just highlight the second entry and press the **INSERT ENTRY** softkey ((F6)), then type "B:". The configuration will now look like this:

```
              Measurement Coprocessor Configuration Utility    B.00.01

 :,1500, 0 A:
 :,1500, 1 B:
 :,1500, 2 C:\HPW
 :,1500, 3 _
 :,1500, 4
```

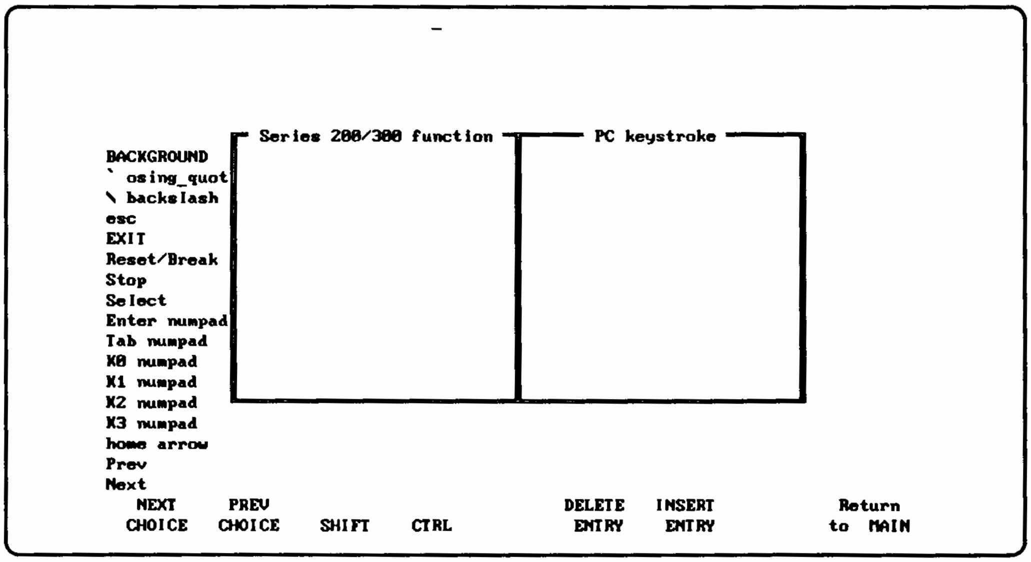
Note that "C:\HPW" will become "drive 2" — and will be accessed at ":,1500,2" — once you store the new configuration.

If you need more help in understanding how to specify MS-DOS path names, refer to chapter 7, "File Systems and Mass Storage." If you have a Vectra RS PC with more than two flexible disk drives, you may also want to read "Configuring Vectra RS Extended Flexible Disk Drives" later in this chapter.

Keydefs

The measurement coprocessor "re-maps" your PC keyboard, using several of its keys to execute HP 9000 Series 200/300 keyboard functions. The "keydefs" field in the configuration screen allows you to redefine the keyboard mapping to meet your own personal preferences.

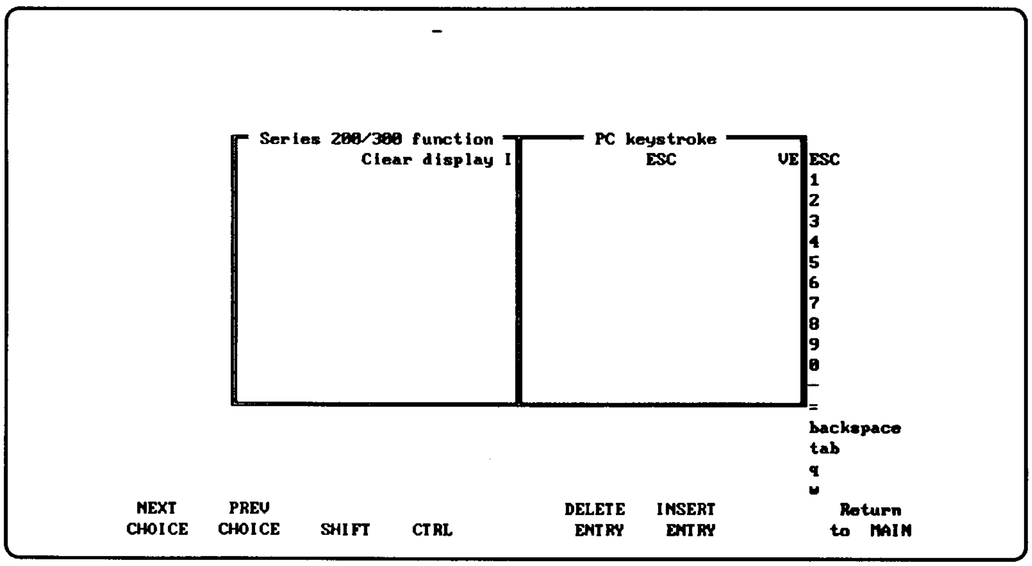
When you highlight the "keydefs" field, the menu changes so that the (F1) key becomes EDIT KEYS. Press EDIT KEYS and a "window" appears in the middle of the configuration screen, as shown below:



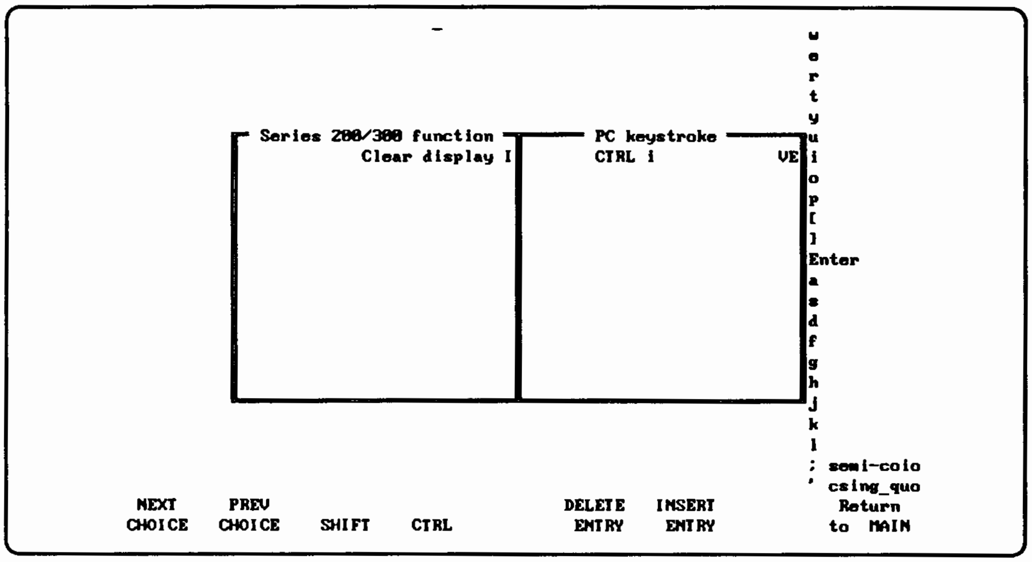
A vertical line divides the window. The left side of the window contains a hidden list of Series 200/300 keyboard functions. The right side of the window contains a hidden list of PC keyboard keys that can be mapped to the functions on the left side. The options are revealed one at a time when you use (F1) (NEXT CHOICE) and (f2) (PREV CHOICE).

As you step through the choices, you will see the key function and, in the right margin, a 1 or 2-letter code. On the left side of the window, the codes will be "N", "I", or both. "N" means the key is found on the Series 200 keyboards, and "I" means the key is found on the the ITF (Series 300) keyboards. On the right side of the window, the codes will be "E", "V", or both. "E" means the key is present on the "Enhanced 101/102-key" (Vectra PC industry-standard) keyboard. "V" means the key is present on the old-style Vectra PC (HP-HIL) keyboard.

An example will clarify the use of "keydefs." Let's assume that you want to map the "Clear Display" function to Ctrl i on the PC keyboard. Press `Next Choice` (F1) several times until "Clear display" appears in the Series 200/300 function window. Then press the "right arrow" cursor key to obtain the following display, which allows you to select a PC keystroke:



Now press `Next Choice` (F1) several times until "i" appears in the PC keystroke window, then press the `CTRL` (F4) softkey (*not* the Ctrl key). The final display appears as follows:



Press `Return to MAIN` (F8) to return to the main configuration screen. Notice that there is a number next to the "keydefs" item. This number corresponds to the number of "keydef" changes you entered. If you save the configuration by pressing `SAVE CONFIG` (F5) and then exit, you will be able to clear the display by pressing Ctrl i whenever you are in BASIC.

Serial Cards

You may configure up to two serial ports (COM1 and COM2) in the configuration file. There are two columns of input spaces next to the list of items. Use one column for each serial port. The default configuration shows "COM1" in the "com port" field in the first column and "NONE" in the second column. To configure "COM2", use the cursor-control keys to move to the "com port" field in the second column and press PREV CHOICE (F2). The selection "COM2" appears along with the default parameters for that port. The screen will appear as follows:

```
             Measurement Coprocessor Configuration Utility   B.00.01

        Primary Configurations                        Serial Cards

DOS cmd save mode: ON                   com port: COM1            COM2
 DOS command wait: ON                   sel code: 9               23
Mouse sensitivity: X=01 Y=00           int level: 6               6
            cache: OFF                      baud: 9600            9600
           drives: 3                      parity: DIS  ODD        DIS  ODD
          keydefs: 1                    char len: 8               8
                                       stop bits: 1               1
                               modem disconnect: CD RI DSR CTS   CD RI DSR CTS

                                                   HP-IB Cards

                                     PC sel code: NONE            NONE
                                    PC int level:
                                        sel code:
                                       int level:

DOS COM Port number this serial card uses

 NEXT     PREV                           SAVE             DEFAULT
CHOICE   CHOICE                         CONFIG             VALUE    EXIT
```

You can change each item in turn by highlighting the appropriate field and using the NEXT CHOICE and PREV CHOICE keys (F1 and F2).

Now let's look at each field in the "Serial Cards" area.

Com Port

The "com port" field allows you to configure one or both of two serial ports, COM1 and COM2. Each port configuration corresponds to the serial interface with the matching switch settings. You may also specify "NONE" for a port, in which case the rest of the settings are not displayed.

If you have connected a serial mouse, you must *unconfigure* the serial port to which it is connected. For example, if you connect a mouse to the COM1 port, you must unconfigure COM1 by setting the "COM1" field to "NONE".

Sel Code

The "sel code" field specifies the select code used by HP BASIC to access the serial interface. The select code value can be either 9 or 23.

Int Level

The "int level" field is used to set the HP BASIC interrupt level for the serial interface. The available values are 3, 4, 5, and 6. However, you should avoid setting 3, which conflicts with select code 15 mass storage. Normally, you should leave the int level set to 6 (the default setting). This interrupt level may be shared with other devices. (Both serial ports can be set to the same HP BASIC interrupt level.)

Baud, Parity, Char Len, and Stop Bits

These four items specify information about communication between HP BASIC and a serial peripheral device (such as a plotter or printer). Refer to the documentation that came with your serial peripheral to determine the proper values.

Modem Disconnect

Each of these fields, when active, disconnects one of the four modem-control lines: CD (carrier detect), RI (ring indicator), DSR (data set ready), and CTS (clear to send). When all four of these fields are "on" (as shown in the previous figure), the four modem-control lines are "disconnected" and are ignored by the serial interface.

The "off" choices are – –, – –, – – –, and – – –. (The number of hyphens matches the number of letters in the "on" position code.) When a field is "off," the corresponding modem-control line is *not disconnected*. Thus, if the fields are set to "– – RI DSR CTS", it means that the CD (carrier detect) line is active and the RI, DSR, and CTS lines are "disconnected."

Refer to your modem documentation to determine what values to use.

HP-IB Cards

You can install up to two HP-IB cards in addition to the measurement coprocessor's built-in HP-IB interface (which is always configured to select code 7). There are two columns of input spaces next to the list of items in the "HP-IB Cards" area. Use one column for each HP-IB card that you want to configure.

The default configuration does not include any HP-IB cards. If you use the cursor-control keys to move to the "NONE" in the first column and then press NEXT CHOICE (F1), the screen will appear as follows:

```
                    Measurement Coprocessor Configuration Utility   B.00.01

          Primary Configurations                          Serial Cards

   DOS cmd save mode: ON                   com port: COM1             COM2
    DOS command wait: ON                   sel code: 9                23
   Mouse sensitivity: X=01 Y=00           int level: 6                6
               cache: OFF                      baud: 9600             9600
              drives: 3                      parity: DIS  ODD         DIS  ODD
             keydefs: 1                    char len: 8                8
                                          stop bits: 1                1
                                   modem disconnect: CD RI DSR CTS    CD RI DSR CTS

                                                          HP-IB Cards

                                        PC sel code: 8                NONE
                                       PC int level: NONE
                                           sel code: 24
                                          int level: 6
```

Each item in the "HP-IB Cards" area is explained below. As you proceed through the items on the screen, read each explanation to find if you need to change the value. Use the function keys to locate the appropriate value for your system setup.

PC Sel Code

The "PC sel code" field is used by the PC emulator to access the HP-IB interface. This is the number you would use to access the interface from MS-DOS. It must match the switch select code set on the HP 82990 HP-IB card. The available values are NONE, and the numbers 1 through 16.

PC Int Level

The "PC int level" field sets the PC interrupt level software configuration — it must match the hardware setting on the HP-IB card. The available values are NONE, 3, 4, 5, and 6. If you enable interrupts, be sure the level you choose does not conflict with any other cards or devices connected to your system. *PC interrupt levels cannot be shared.* Note that COM1 (if available) uses PC interrupt level 4 and COM2 (if available) uses PC interrupt level 3. HP-IB cards do not use interrupts to access mass storage devices, therefore you may select NONE (interrupt disabled) if you intend to use HP-IB to access only mass storage devices.

Sel Code

The "sel code" field is used by HP BASIC to access the HP-IB interface. The value for select code can be either 24 or 25.

Int Level

The "int level" field indicates the HP BASIC interrupt level of the interface. The available values are 3, 4, 5, and 6. However, you should avoid setting 3, which conflicts with select code 15 mass storage. Normally, you should leave the int level set to 6 (the default setting). This interrupt level may be shared with other devices. (Two HP-IB cards can be set to the same HP BASIC interrupt level.)

Exiting the Configuration Utility Program

When you have finished making changes in the configuration utility program, return to the main configuration screen if you haven't already. If you want to save the changes that you have made, press `SAVE CONFIG` (f5), then press `EXIT` (f8) to return to MS-DOS.

If you don't want to store the changes that you have made, don't press `SAVE CONFIG`. Just press `EXIT` and answer "n" (no) when asked if you want to update the disk file. The previous BLP.CON file will be preserved.

Configuring Vectra RS Extended Flexible Disk Drives

The Vectra RS PC provides room for more than two flexible disk drives. Drives "A:" and "B:" are configured for LIF and HPW file access in the usual manner. However, the "extended" (also called "external") flexible disk drives are configured in a slightly different manner.

Note No configuration is necessary for DFS file access to any drive or directory, including the extended flexible disk drives.

The following table shows a typical Vectra RS PC disk drive setup.

MS-DOS Volume Designation	Disk Drive or Volume
A:	5.25-inch flexible disk drive
B:	3.5-inch flexible disk drive
C:	Hard disk volume
D:	Hard disk volume
E:	Hard disk volume
F:	Hard disk volume
G:	3.5-inch flexible disk drive
H:	3.5-inch flexible disk drive

Suppose you want to configure all four flexible disk drives, plus the directory C:\HPW. You might expect that the following configuration would work:

```
                Measurement Coprocessor Configuration Utility   B.00.01

:,1500, 0 A:
:,1500, 1 B:
:,1500, 2 C:\HPW
:,1500, 3 G:
:,1500, 4 H:
```

Actually, this configuration works fine for "A:", "B:", and "C:\HPW". It even allows HPW file access for extended flexible disk drives "G:" and "H:". However, you won't be able to access LIF files in the extended drives. The reason is that for LIF file access, the flexible disk drives are actually accessed using the system BIOS (Basic Input and Output System) drive designations, rather than the MS-DOS designations. The following table again shows the disk drive organization for our "typical" Vectra RS PC, but adds the BIOS flexible disk drive designations.

MS-DOS Volume Designation	BIOS Designation	Disk Drive or Volume
A:	0	5.25-inch flexible disk drive
B:	1	3.5-inch flexible disk drive
C:	-	Hard disk volume
D:	-	Hard disk volume
E:	-	Hard disk volume
F:	-	Hard disk volume
G:	2	3.5-inch flexible disk drive
H:	3	3.5-inch flexible disk drive

The measurement coprocessor software provides a method for you to include the BIOS flexible disk drive designation in your drive specification (for Vectra RS extended flexible disk drives only). In our example, you would enter "2G" instead of "G:" and "3H" instead of "H:" in the configuration window, as shown below:

```
              Measurement Coprocessor Configuration Utility   B.00.01

:,1500, 0 A:
:,1500, 1 B:
:,1500, 2 C:\HPW
:,1500, 3 2G
:,1500, 4 3H
```

This configuration allows LIF file access in all four flexible disk drives, plus the hard disk directory.

Of course, if there were only two hard disk volumes ("C:" and "D:"), the extended flexible disk drives would start at MS-DOS volume "E:", but the BIOS designations would still start at "2". Thus, the correct configuration would be "2E" for the first extended drive, "3F" for the next, and so on. The configuration screen would be as follows.

```
Measurement Coprocessor Configuration Utility   B.00.01

:,1500, 0 A:
:,1500, 1 B:
:,1500, 2 C:\HPW
:,1500, 3 2E
:,1500, 4 3F
```

Additional flexible disk drives would be "4G", and so forth.

Compatibility with Previous-Version Configuration Files

If you have been using an earlier version of the software that used a configuration file named "HPW.CON", you can preserve its configuration for use with your new HP BASIC system.

Version C.00 and later versions of the measurement coprocessor software are *backward compatible* with the "HPW.CON" files used with versions A.00 through A.02. However, you will have to change the name of the file to "BLP.CON". The best way to do this is to use the MS-DOS "COPY" command. For example, if you have an "HPW.CON" file in directory C:\HPW, you could execute the following MS-DOS command:

```
COPY C:\HPW\HPW.CON C:\BLP\BLP.CON  Enter
```

to copy the file to your new system directory (C:\BLP). You can then use CONF.EXE to modify the configuration in the usual manner, or you can use the configuration without modification.

Note

Versions A.00 through A.02 of the software are *not* forward compatible with "BLP.CON" files created or edited with the new CONF.EXE program, regardless of file name.

9

Using Printers and Plotters with the Measurement Coprocessor

A wide variety of printers and plotters are available for personal computers. This chapter tells how to use these printers and plotters with the measurement coprocessor.

HP BASIC Printer and Plotter Commands

HP BASIC includes a number of commands that are useful for dealing with printers and plotters:

- **PRINTER IS** — This command selects a specified device as the default target for the PRINT command.
- **PLOTTER IS** — This command selects a specified device as the default target for all graphics commands.
- **DUMP DEVICE IS** — This command selects a specified device as the default target for DUMP GRAPHICS or DUMP ALPHA commands.
- **PRINTALL IS** — This command selects a specified device as the default target for display logging.
- **OUTPUT** — This command sends strings to a peripheral.

These statements designate the target device with a "device selector," which consists of two parts: an "interface select code" (which designates the interface that the device is connected to) and a "device address" (which designates one of several possible devices that may be connected to that interface).

The measurement coprocessor defines several interface select codes that can be used for printer or plotter access:

Interface Select Code	Interface
7	Measurement coprocessor built-in HP-IB interface
9	PC serial interface (COM1 or COM2)
23	PC serial interface (COM1 or COM2)
24	PC HP-IB card (HP 82335 or HP 82990 HP-IB Card)
25	PC HP-IB card (HP 82335 or HP 82990 HP-IB Card)
26	PC parallel port or network server designated by MS-DOS PRN port

For interfaces that normally connect to only one device at a time (for example, a serial interface), the device selector and the interface select code are one and the same.

However, for interfaces that connect to more than one device at a time (for example, an HP-IB interface), the device select code consists of the two-digit device address appended to the interface select code. For example, to designate a printer at HP-IB address 6 connected to the measurement coprocessor built-in HP-IB interface at interface select code 7, you would execute:

```
PRINTER IS 706
```

either from the BASIC command line, or within a program. It is easy to configure printers and plotters to the measurement coprocessor using these commands. Let's look at some examples.

The HP BASIC statement:

```
DUMP DEVICE IS 26
```

selects a parallel printer as the default DUMP GRAPHICS or DUMP ALPHA device.

The HP BASIC statement:

```
PLOTTER IS 705,"HPGL"
```

selects an HP-IB plotter with address 5, connected to the internal HP-IB interface (select code 7).

The statement:

```
PRINTER IS CRT
```

restores the CRT as the default printer output device.

The statement:

```
PLOTTER IS CRT,"INTERNAL"
```

restores the CRT as the default graphics output device.

As you can see from this quick overview, it isn't difficult to access a printer or plotter from HP BASIC using these commands. (For further information about HP BASIC programming techniques for printers and plotters, refer to *Programming with HP BASIC*.) However, there are some specific details that you need to consider when using printers and plotters with the measurement coprocessor, particularly serial printers and plotters. The rest of this chapter covers these issues.

Configuring HP-IB Printers and Plotters

Configuring HP-IB printers and plotters with the measurement coprocessor is easy. Suppose you have an HP-IB printer (with HP-IB address 6) and an HP-IB plotter (with HP-IB address 5); you can connect them both to the measurement coprocessor's built-in HP-IB interface (using standard HP-IB cables). All of the examples in this section assume this particular configuration.

HP-IB Printers

If you boot and run BASIC, you can use the following program to test printer output:

```
10 PRINTER IS 706
20 FOR N=1 TO 100
30    PRINT "This is a printer-output test!",N
40 NEXT N
50 PRINTER IS CRT
60 END
```

As this program shows, if you execute the BASIC statement:

```
PRINTER IS 706
```

the output of all PRINT statements (as well as the output of BASIC statements like CAT or LIST) goes to the HP-IB printer. As the program also shows, you restore print output to the CRT with:

```
PRINTER IS CRT
```

PRINT ALL Mode

You can also configure BASIC so that *all* display output is sent to the printer, by putting it in PRINT ALL mode. Execute the BASIC statement:

```
PRINTALL IS 706
```

and press "Print All" (the (Print Screen) key on the Vectra PC industry-standard keyboard). This puts the measurement coprocessor in PRINT ALL mode and BASIC responds with:

```
Printall on
```

Everything that is displayed from that moment on is printed as well. If you tire of this, you can press "Print All" again, and BASIC answers with:

```
Printall off
```

Dumping the Alpha and Graphics Screens to a Printer

You can configure BASIC so that BASIC's alpha or graphics display can be dumped to the printer. If you execute the BASIC statement:

```
DUMP DEVICE IS 706
```

and then execute:

```
DUMP ALPHA
```

the text on the display will be dumped to the printer.

Note

The measurement coprocessor normally displays a *combined* alpha/graphics screen on the CRT. However, you may choose to display *separate* alpha and graphics screens (refer to chapter 6). The `DUMP ALPHA` statement always dumps only alpha information — whether in combined mode or separate mode. If you execute `DUMP GRAPHICS` in combined mode, you will get a raster dump of the entire contents of the CRT — both alpha and graphics information will be dumped. In separate mode, `DUMP GRAPHICS` will dump only the graphics screen.

You can use the following program to put graphics on the CRT display:

```
10 PLOTTER IS CRT,"INTERNAL"
20 GCLEAR
30 WINDOW -50,50,-50,50
40 FOR N=50 TO 1 STEP -1
50    MOVE -N,-N
60    DRAW N,-N
70    DRAW N,N
80    DRAW -N,N
90    DRAW -N,-N
100 NEXT N
110 END
```

This program draws 50 concentric rectangles on the display. If you run this program, and then execute the BASIC statements:

```
DUMP DEVICE IS 706
DUMP GRAPHICS
```

the graphics on the display will be dumped to the printer.

HP-IB Plotters

Configuring an HP-IB plotter is just as easy as a printer. You can create a plotter test program by slightly modifying the graphics program of the previous example. Change line 10 to:

```
10 PLOTTER IS 705,"HPGL"
```

and delete line 20. If you run the program, it will draw the fifty concentric rectangles on the plotter instead of the CRT.

Using PC HP-IB Cards

The measurement coprocessor can also access printers and plotters through the HP 82335A or HP 82990A HP-IB Cards. *However, HP-IB cards from other manufacturers are not supported.* The same techniques work, but you must use the correct interface select code (24 or 25).

For example, if you connect an HP-IB printer with address 06 and an HP-IB plotter with address 05 to a PC HP-IB card that has been mapped to select code 24, you would select the printer and plotter as follows:

```
PRINTER IS 2406
PLOTTER IS 2405,"HPGL"
```

Configuring Parallel Printers Through PRN

MS-DOS automatically configures a parallel (Centronics) interface to the PRN (LPT1) device file at power-up. The measurement coprocessor can access the peripheral represented by PRN through interface select code 26, giving the measurement coprocessor access to standard parallel printers.

For example, suppose you have a parallel printer, and you have a Vectra PC with a serial/parallel interface. You can connect the printer to the interface with an HP 24542D cable. You can use the following program to test printer output:

```
10 PRINTER IS 26
20 FOR N=1 TO 100
30    PRINT "This is a printer-output test!",N
40 NEXT N
50 PRINTER IS CRT
60 END
```

Dumping Graphics to a Parallel Printer

There is no problem printing text and dumping the alpha screen to a parallel printer at select code 26. However, there is a complication if you want to do a graphics dump. The problem is that BASIC expects a printer to handle graphics with HP graphics escape sequences, and most PC printers use IBM (Epson) graphics escape sequences. In general, most PC printers do not support HP graphics; and HP printers set to HP graphics mode won't work for PC screen dumps.

This means that if, for example, you have a Hewlett-Packard ThinkJet printer with Centronics parallel interface, you will have to turn-off the ThinkJet's Epson-compatibility switch if you want it to do graphics dumps from the measurement coprocessor.

Configuring Other Devices Through PRN

It is possible to remap the PRN device file to another device by installing a special MS-DOS driver. For example, the HP OfficeShare network allows you to remap PRN to a print server PC that controls a printer. Once this remapping has been done, you can then access the network printer through select code 26, just as if it was a parallel printer.

Configuring Serial Peripherals

Configuring serial printers and plotters is a bit more challenging than configuring a peripheral with an HP-IB or parallel interface because you must go through a series of configuration steps.

The measurement coprocessor allows you to access the COM1 and COM2 PC serial ports through BASIC select codes 9 and 23. You can, for example, select a serial plotter from BASIC with:

```
PLOTTER IS 9, "HPGL"
```

and select a serial printer (on another serial port) with:

```
PRINTER IS 23
```

Note

The measurement coprocessor provides no interrupt support for the PC serial ports, COM1 and COM2. Also, the TRANSFER statement is not supported for COM1 and COM2. Otherwise, the serial STATUS and CONTROL registers are as described in the "Interface Registers" appendix of the HP BASIC *Language Reference* manual.

However, life is not so simple. There are several other things that you must consider in order to successfully communicate with a serial printer or plotter:

1. *Cabling.* Serial interface pinouts are not well standardized from one peripheral to another. You must have a specific cable to interface a specific peripheral to a specific PC serial port.

2. *Line parameters.* A serial communications link between a peripheral and its computer "host" is characterized by a number of parameters. *Note that both the host and the peripheral must be set to the same line parameters to make communications possible. The host can be configured through software. The peripheral is generally configured by setting DIP switches.*

   - Transmission speed, or "baud rate", in bits per second — typical values are 300, 1200, 2400, and 9600 bits per second.
   - Character length — a serial port can send data 5, 6, 7, or 8 bits at a time (though in general nobody uses anything but 7 or 8 bits).
   - Parity — an error checking bit can be optionally added to the data bits sent. This "parity" bit can be:

     Always zero.

     Always one.

     Odd Parity. (Selected to make the number of "1" data bits always equal an odd number.)

     Even Parity. (Selected to make the number of "1" data bits always equal an even number.)
   - Stop bits — the transmitted character can be terminated with an additional 1, 1.5, or 2 "stop bits" (to separate it from the next character coming across the link).

3. Handshaking. Printers and plotters print or plot far more slowly than they can receive data from the host; while most peripherals can store a certain amount of input data in their local "buffer" memories, if the peripheral has no means of telling the host to stop sending data, the host will continue to send data when the buffer is full — and that data will be lost. Fortunately, a serial link has control lines that allow the peripheral to signal to the host that its buffer is full — a procedure known as "handshaking".

4. Plotter configuration. HP plotters are programmable; they accept HPGL (Hewlett-Packard Graphics Language) graphics commands as well as configuration commands. Some configuration commands must be executed on a serial plotter before the plotter can be designated as the plot output device with PLOTTER IS.

The following sections explain each of these issues in detail.

Cabling

Your HP Vectra PC may be equipped with a dual-serial card or with a serial/parallel card. The dual-serial card contains two serial ports, one with a 9-pin connector, the other with a 25-pin connector. The serial-parallel card contains a parallel port with a 25-pin connector and a serial port with a 9-pin connector. The 9-pin serial port has exactly the same configuration for both cards.

The 9-pin port on both cards links to virtually all HP serial printers and plotters through an HP 24542G serial cable. The sole exception is the HP 7550 Plotter, which requires an HP 24542H cable.

The 25-pin port on the dual-serial card links to virtually all HP serial printers and plotters through an HP 13242G serial cable. Again, the sole exception is the HP 7550 plotter, which requires an HP 17255D cable.

If you are using non-HP serial cards and peripherals, you must select your cable carefully based the pinout information given in the manuals that came with your equipment.

Line Parameters

Setting the proper line parameters for the measurement coprocessor is easy using the configuration utility "CONF.EXE". This utility allows you to set the following serial port parameters for up to two PC serial ports:

- DOS COM port (COM1 or COM2).
- Corresponding BASIC select code (9 or 23).
- Baud rate (50 through 19200 bps, in a number of steps).
- Parity (off or on; zero, one, odd, even).
- Character length (5, 6, 7, or 8 bits).
- Stop bits (1, 1.5, or 2).
- Modem disconnect.

Refer to chapter 8 for a complete description of CONF.EXE and the selection of line parameters.

Handshaking

Handshaking can be implemented with the measurement coprocessor using the DSR serial control line. This line can be used by a peripheral to tell the measurement coprocessor that it is ready for more data (by asserting DSR) or not ready for more data (by releasing DSR). You can set up DSR-line handshaking for the measurement coprocessor by using the CONF.EXE utility. Just clear the modem disconnect parameter's "DSR" field, as described in chapter 8. After the next RESET, handshaking will occur automatically.

Plotter Configuration Commands

Configuring a serial plotter has an added complication over configuring a serial printer. Not only must the host's serial interface be programmed, but the plotter's interface must be programmed as well. For an HP plotter the process is easy. You simply send the plotter commands as a string of ASCII characters using the HP BASIC OUTPUT statement. The syntax is:

`OUTPUT` *select code*; "*string*"

For example, to send the "Store Pen" (SP) command to a serial plotter at select code 9, use the statement:

```
OUTPUT 9; "SP;"
```

A complete discussion of plotter commands is beyond the scope of this manual. However, there are two plotter commands that you will need to use in order to successfully control a plotter from the measurement coprocessor. These are the "Set Output Mode" and "Set Extended Output and Handshake Mode" commands, which set the following parameters:

- *Turnaround delay time.* This parameter specifies the amount of time the plotter will delay before responding to a status query from BASIC. If no delay time is specified, the plotter will respond immediately, before BASIC is ready.

- *Output terminator characters.* When the plotter sends a response to the host, the response is terminated with the specified output terminator characters. BASIC expects plotter responses to end with a carriage-return/line-feed sequence (ASCII decimal codes 13 and 10). Thus, the plotter must be programmed to end its responses with these two characters.
- *Inter-character delay.* This parameter specifies the amount of time between characters in the plotters response to the host. In most cases BASIC can keep up with the plotter's maximum transmission rate, but you can include a "safety factor" by programming an inter-character delay.

The "Set Output Mode" command has the following syntax:

`Esc.M` *turnaround delay*;;;*output terminator*;*output terminator*:

Where "Esc" is the Escape character (ASCII decimal code 27). The turnaround delay time is specified in milliseconds, and must be less than 54,612 milliseconds. The two output terminator characters are specified by their decimal codes.

From HP BASIC, you can send the "Set Output Mode" command using an OUTPUT statement. For example, to specify a turnaround time of 250 milliseconds and carriage-return/line-feed as the termination characters, execute the following BASIC statement:

```
OUTPUT 9; CHR$(27)&".M250;;;13;10:"
```

The "Set Extended Output and Handshake Mode" command has the syntax:

`Esc.N` *inter-character delay*;:

Again, you can use an OUTPUT statement to send the command to the plotter. For example, to set an inter-character delay of 10 milliseconds, execute the following BASIC statement:

```
OUTPUT 9; CHR$(27)&".N10;:"
```

10

Sharing Files with MS-DOS Applications

HP BASIC is an ideal programming environment for instrument control and data acquisition. On the other hand, you may have a favorite MS-DOS application program that you use for data analysis. Also, the HP BASIC editor gives you very powerful program editing capabilities. But there are some circumstances where you may want to use an MS-DOS editor to edit your BASIC programs. In the past, it has been somewhat inconvenient to transfer data and program files from HP BASIC to an MS-DOS application. Now, however, the measurement coprocessor DFS mass storage system allows you to create files from HP BASIC, store data in them, and then directly access those files from an MS-DOS application. You can also list a program to an MS-DOS file, and access it with an MS-DOS editor. This chapter explains how to use these capabilities, and gives some practical examples.

Sharing Data with an MS-DOS Spreadsheet Program

A very popular feature of the measurement coprocessor is the capability to collect data with a BASIC program and then store the data in a file that is directly readable by an MS-DOS spreadsheet program (for example, Lotus 1-2-3). The DFS file system makes this possible.

Note

Only the *DFS* mass storage system uses the MS-DOS internal file structure. LIF and HPW files, regardless of file type, are incompatible with MS-DOS. Thus, you should use the DFS file system if you want to share files with an MS-DOS application.

Within the DFS file system, there are three types of data files: "DOS", "ASCII", and "BDAT". These file types are covered in detail in chapter 6 of *Programming with HP BASIC*. For our purposes here, remember that ASCII and BDAT files in the DFS file system have "headers" that identify the size and type of file. Thus, although these files are MS-DOS formatted, they cannot be read directly from an MS-DOS application. On the other hand, a file of the "DOS" file type has no header. An HP BASIC "DOS" type file is quite simply an MS-DOS text file, which exists in an MS-DOS directory. Thus, an HP BASIC "DOS" type data file can be read directly by an MS-DOS application program, provided the data is presented in the form expected by the application.

Most applications expect some sort of *delimiter* (item separator) between data items. The space (ASCII character number 32) is often used — or a carriage-return/line-feed (ASCII characters 13 and 10, respectively). You will need to determine the required data format for your MS-DOS application. The example in the following section illustrates the process for Lotus 1-2-3.

A Lotus 1-2-3 Example

You can use the following program to control an HP 3457A Multimeter. The meter will take a reading approximately every second for 50 seconds. The 50 readings are output to a DOS type file with the data delimited by spaces to separate the items in a row, but with each row terminated by a carriage-return/line-feed. This file is set up to be read by Lotus 1-2-3, but you can modify the program to delimit the data properly for other MS-DOS applications.

```
10    ! This program controls an HP 3457A Multimeter at address 722.
20    !
30    CLEAR 722                         ! Clear the meter
40    OUTPUT 722; "PRESET"              ! Put meter in PRESET mode:
50                                      !    DC VOLTS (autoranging)
60                                      !    ASCII data output
70                                      !    Synchronous triggering
80    !
90    MASS STORAGE IS "\BLP:DOS,C"
100   DIM A$[128]                               ! Dimensions a string.
110   PURGE "DATAFILE.PRN"                      ! Discards previous version.
120   CREATE "DATAFILE.PRN",1                   ! Creates new DOS data file.
130   ASSIGN @File TO "DATAFILE.PRN";FORMAT ON  ! Assigns I/O path to file.
140   !
150   FOR I=1 TO 50
160     WAIT 1                          ! Takes reading every second.
170     ENTER 722;A                     ! Enters reading from DVM.
180     DISP A                          ! Displays reading.
190     IF I MOD 8=0 THEN
200       A$=VAL$(A)&CHR$(13)&CHR$(10)  ! Every 8th reading followed by CR/LF.
210     ELSE
220       A$=VAL$(A)&CHR$(32)           ! Other readings followed by a space.
230     END IF
240     OUTPUT @File USING "#,K";A$     ! Outputs data point as string to file.
250   NEXT I
260   !
270   ASSIGN @File TO *                 ! Closes I/O path.
280   END
```

Statements 30 and 40 clear the meter and put it into its PRESET mode. In PRESET mode the meter takes a DC voltage reading every time the meter output buffer is read. (This is called "synchronous triggering.")

Statements 100 through 130 dimension a string and set up the data file named "DATAFILE.PRN". Note that this program assumes that a previous version of the file exists before you run the program. Thus, the PURGE statement of line 110 is included to remove it.

The FOR ... NEXT loop (lines 150 through 250) performs several steps repeatedly, once for each of 50 readings from the multimeter. Line 170 enters a reading, which is displayed on the CRT by line 180. For every eighth reading line 200 converts the value of variable A (the reading) to a string, with a carriage return (ASCII character number 13) and line feed (ASCII 10) appended. For all other readings line 220 does the same thing, but appends a space (ASCII character number 32) instead. Each reading, in its turn, is output to the data file by the OUTPUT statement in line 240. Each reading has either a space or a carriage return/line feed appended.

The resulting data file contains 50 readings arranged into rows and columns. There are eight columns since every eighth reading is followed by a carriage return/line feed. If you exit HP BASIC and execute the following MS-DOS command:

```
PRINT C:\BLP\DATAFILE.PRN  Enter
```

the file will be printed to the system printer (PRN). The following is a typical printout (the values are in volts):

```
.962182 .962219 .962216 .962107 .962052 .962126 .962231 .962778
.964213 .965314 .966228 .966878 .967415 .968132 .968718 .969352
.969904 .97019 .970613 .9711 .971406 .971999 .972497 .972807
.973276 .973836 .974384 .974995 .97578 .976093 .976431 .97682
.97693 .977023 .976799 .976593 .976682 .976766 .976929 .976971
.97717 .977407 .977626 .977886 .978339 .978817 .979155 .97959
.979974 .980173
```

When you import this file into Lotus 1-2-3, the rows and columns in the file become rows and columns in the Lotus 1-2-3 spreadsheet. Each reading occupies one "cell" in the spreadsheet.

Data Collection in Background

You can write an HP BASIC program that puts the measurement coprocessor into background mode and then continues to perform a task while you have full use of your PC from MS-DOS. For example, you might want write a program that takes a series of measurements in background mode while you run an MS-DOS application in foreground.

The following is a modification of the previous example. Again, 50 readings are taken and put into a DOS file, but in this case the readings are taken in background mode. Again, the resulting DOS file can be read from Lotus 1-2-3.

```
10    ! This program controls an HP 3457A Multimeter at address 722.
20    !     THIS VERSION RUNS WITH BASIC IN BACKGROUND MODE.
30    CLEAR 722                           ! Clear the meter
40    OUTPUT 722; "PRESET"                ! Put meter in PRESET mode:
50                                        !    DC VOLTS (autoranging)
60                                        !    ASCII data output
70                                        !    Synchronous triggering
80    !
90    MASS STORAGE IS "\BLP:DOS,C"
100   DIM A$[128]                                 ! Dimensions a string.
110   PURGE "DATAFILE.PRN"                        ! Discards previous versios.
120   CREATE "DATAFILE.PRN",1                     ! Creates new DOS data file.
130   ASSIGN @File TO "DATAFILE.PRN";FORMAT ON    ! Assigns I/O path to file.
132   !
134   CONTROL @File,9;0                 ! SPECIFIES UNBUFFERED OUTPUT TO FILE.
136   OUTPUT 19; "BACKGROUND"           ! PUTS BASIC IN BACKGROUND MDOE.
140   !
150   FOR I=1 TO 50
160     WAIT 1                            ! Takes reading every second.
170     ENTER 722;A                       ! Enters reading from DVM.
180     DISP A                            ! Displays reading.
190     IF I MOD 8=0 THEN
200       A$=VAL$(A)&CHR$(13)&CHR$(10)    ! Every 8th reading followed by CR/LF.
210     ELSE
220       A$=VAL$(A)&CHR$(32)             ! Other readings followed by a space.
230     END IF
240     OUTPUT @File USING "#,K";A$       ! Outputs data point as string to file.
250   NEXT I
260   BEEP 400,1                         ! BEEPS WHEN DONE TAKING READINGS.
270   ASSIGN @File TO *                   ! Closes I/O path.
280   END
```

The following changes have been made to the previous program:

- The CONTROL statement in line 134 specifies *unbuffered* output of data through the I/O path to ensure no data is lost when the file is read.
- The statement OUTPUT 19; "BACKGROUND" in line 136 puts the measurement coprocessor into background mode before the FOR ... NEXT loop begins.
- The statement BEEP 400,1 (line 260) causes the computer to beep at approximately 400 Hz for one second when the FOR ... NEXT loop is finished. (When all 50 readings have been taken.)

Like the previous program, this program takes 50 readings from the HP 3457A Multimeter, converts each reading to a string, and outputs the string (with a space or CR/LF appended) to a DOS type data file. The data file consists of rows and columns and can easily be imported into Lotus 1-2-3. The difference here is that the program runs with the measurement coprocessor in background, which allows you to run an MS-DOS application in foreground while the readings are being taken. Although you may not gain much while taking only 50 readings, suppose you had written the program to take 500 readings. By using background mode you free the computer for other use while the readings are being taken.

One problem is that, although an MS-DOS application can easily read the DOS file "DATAFILE.PRN", you must avoid accessing the file until the BASIC program is done doing so. That is the purpose of the distinctive beep sound produced by line 170. The beep tells you that the program is done taking readings and accessing the data file. After the beep, you can access the file safely from MS-DOS.

Although this program allows you to take readings in background while using the computer for MS-DOS applications, and although the program produces a data file readable from MS-DOS, you can take this process a step farther. Your measurement coprocessor software includes utilities that allow you to *integrate* HP BASIC programs with MS-DOS applications. Instead of just beeping to indicate that it is done, your HP BASIC program can cause a "pop-up window" to appear in your MS-DOS application program screen. The window can give you information or request input from you. You can also use these utilities to bring data from HP BASIC into your MS-DOS application while it is running. For detailed information on this subject, refer to *Integrating HP BASIC with MS-DOS Applications.*

Editing Programs with an MS-DOS Editor

Sometimes you may want to use your favorite MS-DOS editor (for example, Executive MemoMaker or Microsoft Word) to edit an HP BASIC program. For example, suppose you have a long BASIC program with several WAIT statements. You may want to use your MS-DOS editor to search for "WAIT 1" and replace it with "WAIT 3". Or perhaps your program has a lot of screen messages. Although HP BASIC will check syntax interactively, it won't check your spelling in those messages. If you could export your program listing to MS-DOS, you could use an MS-DOS editor to check spelling. Well, it's really quite easy to do this. All you have to do is create a DOS type file in a DFS directory, and list the program to that file.

For example, to list the BASIC program "OLDPROG.RMB" (a PROG file in the directory "C:\BLP") into an MS-DOS file, just execute the following commands from the BASIC command line:

`MSI "\BLP:DOS,C"`	*Make C:\BLP the current mass storage.*
`LOAD "OLDPROG.RMB"`	*Load the BASIC program.*
`CREATE "PROGLIST.TXT",1`	*Create the listing file.*
`PRINTER IS "PROGLIST.TXT"`	*Redirect printer output to the file.*
`LIST`	*List the program into the file.*
`PRINTER IS CRT`	*Restore printer output to the CRT.*

The file "PROGLIST.TXT" is actually an MS-DOS text file with each program line terminated with a carriage-return/line-feed. Now you can use your MS-DOS editor to edit the program. You can even merge the listing into another text file — to include your listing in a report, for example. (Actually, many of the examples in this manual were imported into the chapter files in exactly this way.)

One word of caution. The MS-DOS editor will allow you to do *anything* to your program listing — there is no syntax checking. However, don't worry too much about introducing errors. When you return your program to BASIC, anything illegal will be commented out. You will get an error message when you run the program, and you should have no problem debugging the program with the BASIC editor.

To return our example program to HP BASIC, execute the following statement:

```
GET "PROGLIST.TXT"
```

Run the program and debug it if necessary. Once you are satisfied, you can STORE the program as follows:

```
STORE "NEWPROG.RMB"
```

BASIC ASCII to MS-DOS Text File Conversions

For your convenience, an HP BASIC program named "ASCI2DOS" is included on your "Manual Examples, LIFINIT, and Selected CSUBs" disk. You can use this program to convert any HP BASIC "ASCII" file into an MS-DOS text file. You can also convert an MS-DOS text file into an HP BASIC "ASCII" file. To load this program just insert the "Manual Examples, LIFINIT, and Selected CSUBs" disk in drive A: and execute:

```
LOAD "ASCI2DOS:DOS,A"
```

The program is user-interactive. Just press RUN to start the program and then follow the instructions displayed on the screen.

11

Multiple Coprocessors and Special Configurations

This chapter covers several advanced topics. First, we'll look at how to configure and run two or three measurement coprocessors at the same time. Second, we'll consider how to create multiple configurations for running one or more measurement coprocessors. Third, we'll cover how to install the measurement coprocessor software manually. Finally, we'll cover the syntax of the "DEVICE=HPBLP.SYS" statement and how to resolve interrupt and address conflicts.

Multiple Measurement Coprocessors

You can install up to three measurement coprocessors in your PC. You can install either HP 82300C Measurement Coprocessors or HP 82324A High-Performance Measurement Coprocessors, or a combination of both types, *provided each one is configured to a different "card" configuration.*

Configuring Multiple Coprocessors

As mentioned previously, the measurement coprocessor software provides predefined software configurations for up to three measurement coprocessors. These software configurations are as follows:

- "Card 1": I/O address 250h and interrupt level (IRQ) 7.
- "Card 2": I/O address 280h and interrupt level (IRQ) 9.
- "Card 3": I/O address 330h and interrupt level (IRQ) 5.

You can change the predefined software configurations, as described later in this chapter. However, unless you encounter an address or interrupt conflict with some other PC card, it is recommended that you use the standard "card" configurations. All you need to do is set the hardware configuration of each measurement coprocessor to match a different software "card" configuration:

- For a HP 82300C Measurement Coprocessor, set the I/O address and interrupt switches to match one of the software configurations. The factory default is "Card 1" (250h and IRQ7).
- For a HP 82324A High-Performance Measurement Coprocessor, just set the I/O address switches. The interrupt level is self configuring. Again, the factory default is "Card 1".
- If you have both an HP 82300C and an HP 82324A, leave the HP 82300C in the card 1 (factory default) settings and change the HP 82324A. This will save you time because it is easier to change the switches on the HP 82324A.

Note If two measurement coprocessors have the same I/O address or interrupt level, the conflict will result in an error. If you install a measurement coprocessor with a hardware configuration that doesn't match one of the software "card" configurations, the measurement coprocessor software won't be able to find it.

Running Multiple Coprocessors

Chapter 3 covered the procedure for booting or restarting a specific measurement coprocessor. By way of review, from the MS-DOS command line:

`BASIC 1` (Enter)	*boots (or restarts) Card 1.*
`BASIC 2` (Enter)	*boots (or restarts) Card 2.*
`BASIC 3` (Enter)	*boots (or restarts) Card 3.*

You can supply the usual boot options after the number. For example:

```
BASIC 3 /BOOT  (Enter)
BASIC 1 /BOOT /MENU  (Enter)
BASIC 2 /BOOT SYSBA514  (Enter)
BASIC 3 /BOOT \BLP\SYSBA514  (Enter)
```

All of the boot options have the usual meaning, but the specified measurement coprocessor is rebooted.

A few points need to be considered:

1. If you specify a non-existent measurement coprocessor, an error results.
2. If you leave out the measurement coprocessor "card" number, the lowest numbered measurement coprocessor present will be booted (or restarted).
3. If you specify the system file name to boot ("SYSBA514") and that file is not in the current directory, you will need to give the path to the file.
4. If a measurement coprocessor has Series 200 boot ROMs, you will need to follow the special instructions provided for booting with those ROMs. Refer to "Booting with Series 200 Version 3.0 Boot ROMs" in chapter 3.

Let's suppose that you have three measurement coprocessors installed. Each one is configured to match one of the three standard "card" configurations. Card 1 is an HP 82300C Measurement Coprocessor, Card 2 is an HP 82300B BASIC Language Processor (with Series 200 boot ROMs), and Card 3 is an HP 82324A High-Performance Measurement Coprocessor. When you boot your PC, the following will appear at the end of the boot sequence:

```
Hewlett-Packard Measurement Coprocessor Device Driver Version A.01.00
Copyright (c) Hewlett-Packard Co. 1989, 1990.  All rights reserved.
      Card  Type    Addr  IRQ              Card State
      ----  -----   ----  ---   ----------------------------------
       1    82300   250H   7     Ready to boot (PC300 BOOTROM 1.0)
       2    82300   280H   9     Ready to boot (S200  BOOTROM 3.0)
       3    82324   330H   5     Ready to load boot code
```

This screen shows that the three measurement coprocessors ("Card 1","Card 2", and "Card 3") have the standard address and interrupt-level configurations, and that all three are ready to boot. Card 2 is identified as having the "old" boot ROMs, so you will have to follow the "hit spacebar" procedure if that coprocessor has access to a system file on an HP-IB disk drive, an SRM network, or in ROM (refer to chapter 3).

So far we've covered how to boot an *individual* measurement coprocessor. However, you can run HP BASIC on up to three measurement coprocessors at the same time. All three can run HP BASIC in *background* mode (refer to chapter 4), but only one measurement coprocessor can run HP BASIC in *foreground* mode at a given time.

Note

The software installation procedure installs just one system on your hard disk (normally in the directory "C:\BLP"). You can use this system to boot all three measurement coprocessors. In fact, *you should not attempt to install more than one system on your hard disk.* If you want to use a different configuration for each measurement coprocessor, refer to "Creating Multiple Configurations" later in this chapter.

Let's boot all three measurement coprocessors. You can boot BASIC on Card 1 by typing:

```
BASIC 1 [Enter]
```

from the MS-DOS command line. You can then start a program and press [Ctrl] [F9] to put Card 1 in background. Or you can execute the statement:

```
OUTPUT 19; "BACKGROUND"
```

within your program.

Once Card 1 is in background, you can execute the following command from MS-DOS:

```
BASIC 2 /BOOT \BLP\SYSBA514 [Enter]
```

to boot Card 2, which has Series 200 boot ROMs. Now you can start a program on Card 2, and then put Card 2 in background. Once Card 2 is in background, you can boot Card 3 as follows:

```
BASIC 3 [Enter]
```

Thus, all three measurement coprocessors are now running BASIC at the same time (Card 1 and Card 2 in background, and Card 3 in foreground). You can start a program on Card 3 and then put it into background too.

There are some restrictions on running programs in background. In particular, you have to be careful when accessing mass storage from background. These restrictions are covered in the section "Background Mode and the Measurement Coprocessor" in chapter 4. When you want to terminate a background process, return the measurement coprocessor to foreground mode the same way you normally start BASIC. For example, type:

```
BASIC 2  (Enter)
```

from the MS-DOS command line to return Card 2 to foreground mode. You can then exit BASIC normally by pressing (Ctrl) (F10).

Multiple Coprocessor Applications

You can run HP BASIC programs in background on up to three measurement coprocessors at the same time, as noted earlier. If each measurement coprocessor program is running independently, you need only be concerned about the restrictions on background operation covered in chapter 4 and the limitations described in "Memory Requirements", below. However, to obtain the full versatility of a multiple coprocessor system, you can establish a "smart" communication channel between your measurement coprocessors and MS-DOS, and between individual coprocessors. The measurement coprocessor software includes utilities to enable you to do this. Refer to the manual *Integrating HP BASIC with MS-DOS Applications* for detailed information.

Memory Requirements

The main BASIC program (BASIC.EXE), which controls the measurement coprocessor in foreground mode, occupies approximately 295 KB of PC memory (user RAM). When you put the measurement coprocessor into background, this memory is freed, but a memory resident program, which occupies approximately 100 KB of PC memory, is loaded. In fact, a memory resident program is loaded for *each* measurement coprocessor that is running in background mode.

If you have two measurement coprocessors running in background, and one in foreground, most of your PC memory will be occupied. On the other hand, three measurement coprocessors running in background occupy a total of about 300 KB of PC memory, leaving approximately half of your PC memory available for MS-DOS applications. This memory requirement may place restrictions on the MS-DOS applications that you can run in foreground.

Creating Multiple Configurations

As mentioned previously, the installation procedure only installs one HP BASIC system on your hard disk. *You should not attempt to install more than one system.* However, you can use this system to boot up to three measurement coprocessors.

In some cases, you may want to use a different configuration for each measurement coprocessor. Or you may want to have an alternate configuration for a single coprocessor. The following sections show how you can do this.

Multiple Configurations for One Coprocessor

When you installed your measurement coprocessor software, the INSTALL program installed the configuration utility "CONF.EXE" and a default configuration file "BLP.CON" in your system directory (normally "C:\BLP"). Chapter 8 tells how to run the "CONF.EXE" utility and customize your "BLP.CON" file.

If you want to create more than one bootable configuration, you can create several configuration files, each named "BLP.CON", and each in a different directory. For example, if you have completed the installation in C:\BLP, you can create a second bootable configuration as follows:

1. Create a new directory with the MS-DOS "MKDIR" command. For example:

   `MKDIR C:\BLP-2` (Enter)

2. Copy the "BLP.CON" file and "CONF.EXE" utility to the new directory with the MS-DOS "COPY" command:

   `COPY C:\BLP\BLP.CON C:\BLP-2` (Enter)

   `COPY C:\BLP\CONF.EXE C:\BLP-2` (Enter)

3. Change to the new directory:

   `CD C:\BLP-2` (Enter)

4. Run the "CONF.EXE" utility *in the new directory* to modify the configuration in the new "BLP.CON" file. Execute:

   `CONF` (Enter)

5. Following the directions in chapter 8, modify "BLP.CON" in the new directory, save the new configuration, and exit "CONF.EXE".

Once you have created the alternate configuration, you have two options. To boot HP BASIC with the alternate configuration, change directories to C:\BLP-2 and then boot BASIC:

```
CD C:\BLP-2  [Enter]

BASIC /BOOT  [Enter]
```

To boot HP BASIC with the regular configuration ("BLP.CON" in the "C:\BLP" directory), boot BASIC *from any other directory.* For example:

```
CD C:\  [Enter]

BASIC /BOOT  [Enter]
```

Note In order to invoke the new configuration, you must *boot* the measurement coprocessor. If the measurement coprocessor is already booted, you must reboot it with the "/BOOT" option, as shown above.

Multiple Configurations for Multiple Coprocessors

Now let's consider a multiple coprocessor example. Suppose that you have three measurement coprocessors installed and you want to use a different configuration for each. You could set up three configurations as follows:

1. Install the measurement coprocessor software in directory "C:\BLP" and modify "BLP.CON" to the desired configuration for Card 1.
2. Copy "BLP.CON" and "CONF.EXE" to a second directory "C:\BLP-2". Using "CONF.EXE", modify that "BLP.CON" file to the desired configuration for Card 2.
3. Copy "BLP.CON" and "CONF.EXE" to a third directory "C:\BLP-3". Using "CONF.EXE", modify that "BLP.CON" file to the desired configuration for Card 3.

To start Card 1, change directories to any directory *except* C:\BLP-2 or C:\BLP-3 and then boot HP BASIC. The "BLP.CON" configuration from C:\BLP will be used.

To start Card 2, change directories to C:\BLP-2 and then boot HP BASIC. The "BLP.CON" configuration from that directory will be used.

To start Card 3, change directories to C:\BLP-3 and then boot HP BASIC. The "BLP.CON" configuration from that directory will be used.

Installing HP BASIC Manually

The INSTALL program provides a fully automatic installation of your measurement coprocessor software, which is described in chapter 2. As part of the installation, the program makes some changes to your "CONFIG.SYS" and "AUTOEXEC.BAT" files. (These files are found, or will be created, in the root directory: "C:\".) If you prefer to make the changes to "CONFIG.SYS" and "AUTOEXEC.BAT" manually, you can use the manual installation procedure that follows.

The following sections describe the changes that the INSTALL program makes in your "CONFIG.SYS" and "AUTOEXEC.BAT" files, and how you can make these changes manually. Actually, the INSTALL program still does most of the work automatically, but you'll need to edit the "CONFIG.SYS" and "AUTOEXEC.BAT" files using an MS-DOS editor. You will also need to copy the driver program "HPBLP.SYS" to the hard disk manually.

Using an MS-DOS Editor

You can use any convenient MS-DOS editor program capable of storing an MS-DOS ASCII file to modify "CONFIG.SYS" and "AUTOEXEC.BAT". For example, you could use Executive MemoMaker or Microsoft Word — just make sure that you store the file as an *unformatted* MS-DOS ASCII file. Refer to the user's manual that came with your editor for instructions. If you do not have such an editor, you can use EDLIN to edit your files. Refer to your MS-DOS documentation for instructions on using EDLIN.

Modifying CONFIG.SYS

Using your MS-DOS editor, make the following changes to "CONFIG.SYS":

1. Add the lines:

   ```
   BUFFERS=20
   ```

   and

   ```
   FILES=20
   ```

   (If "CONFIG.SYS" already has BUFFERS and FILES specifications, make sure that they are both at least equal to 20. If they are larger than 20, you need not change them.)

2. Add the line:

   ```
   DEVICE=HPBLP.SYS
   ```

The following is a typical CONFIG.SYS file that you could use with HP BASIC:

```
BUFFERS = 20
FILES = 20
DEVICE=HPBLP.SYS
```

Note

In some instances you may need to have more than 20 files open. You can set FILES to a higher value, but this uses up additional memory.

You can specify different options as parameters in the DEVICE=HPBLP.SYS statement. Refer to "Device Driver Statement Syntax with Options" for the complete syntax of this statement.

Modifying AUTOEXEC.BAT

Using your MS-DOS editor, make the following changes to AUTOEXEC.BAT. *The changes below assume that you are installing your BASIC system in the directory "C:\BLP". If not, substitute the name of the directory that you are using in each place that "C:\BLP" appears.*

1. Add the following lines at the beginning of your file. (The automatic installation also puts the line "C:" at the top of the file, but this line is not necessary for the manual installation.) *Type these lines exactly as shown. The commands are case sensitive.*

   ```
   set blppath=C:\BLP
   set BLPDIR1=.;C:\BLP
   set BLPDIR2=.;C:\BLP
   set BLPDIR3=.;C:\BLP
   ```

2. Add the following line immediately after the MS-DOS PATH command. (If there is no PATH command, add the "set path" command right after the "set BLPDIR3" command.)

   ```
   set path=%path%;%blppath%
   ```

The following is a typical "AUTOEXEC.BAT" file that you could use with HP BASIC:

```
set blppath=C:\BLP
set BLPDIR1=.;C:\BLP
set BLPDIR2=.;C:\BLP
set BLPDIR3=.;C:\BLP
PATH=C:\;C:\DOS
set path=%path%;%blppath%
PROMPT=$P$G
```

Copying the Driver

The driver program "HPBLP.SYS" is provided on Measurement Coprocessor Software disk 1. You can copy this program to the hard disk using the MS-DOS "COPY" command. Follow these steps:

1. Insert disk 1 (either 3.5-inch or 5.25-inch, whichever is appropriate) in drive A:.
2. Type the following from the MS-DOS command line:

   ```
   COPY A:\HPBLP.SYS C:\  (Enter)
   ```

3. Remove disk 1.

Completing the Installation

Once you have modified CONFIG.SYS and AUTOEXEC.BAT, and you have copied the driver, you can complete the installation with the INSTALL program. First, reboot the PC by pressing the (Ctrl), (Alt), and (Del) keys simultaneously. Once the PC finishes booting, re-insert disk 1 in drive A: and type:

```
A:  (Enter)
```

from the MS-DOS command line. This changes the current drive to "A:". To start the INSTALL program type:

```
INSTALL  (Enter)
```

The INSTALL program will proceed automatically with the rest of the installation. Just follow the instructions on the screen. The procedure is the same as that given in chapter 2, except that the PC will not be rebooted.

Device Driver Statement Syntax with Options

As mentioned earlier in this chapter, the device driver "HPBLP.SYS" must be declared in a "DEVICE=HPBLP.SYS" statement in your "CONFIG.SYS" file. If you want to use only the default parameters, the statement is:

```
DEVICE=HPBLP.SYS
```

However, the complete syntax of this statement is:

DEVICE=HPBLP.SYS [/1:*int*[:*ioaddr*:[*bufseg*]] /2:*int*:[*ioaddr*:[*bufseg*]] /3:*int*:[*ioaddr*:[*bufseg*]]]

where:

1, 2, or 3	Specifies the "card" number of each measurement coprocessor.
int	Specifies the interrupt level (IRQ) for the specified "card." The allowed values are: HP 82300: 3, 4, 5, 7, and 9. (Value must match the IRQ jumper setting on the measurement coprocessor.) HP 82324: 3, 5, 7, 9, 10, 11, 12, and 15. (The high-performance measurement coprocessor auto-configures to the interrupt level selected. Note that interrupt level 4 is *not* supported.)
ioaddr	Specifies the I/O address of the specified "card." The allowed values are: HP 82300: 250, 280, 330, and 390. The hexadecimal address value must match the "primary address" switch setting on the measurement coprocessor. HP 82324: 250, 280, 330, and 390. The hexadecimal address value must match the "primary address" switch setting on the high-performance measurement coprocessor *unless secondary addressing is used.* By using secondary addressing, additional I/O address

values are possible. For further information, refer to "Using Secondary Addressing (HP 82324A Only)" later in this chapter.

bufseg Specifies the segment value of the buffer address used for loading the boot code into the HP 82324A High-Performance Measurement Coprocessor. This buffer address value has no effect on the HP 82300C Measurement Coprocessor. For further information, refer to "Resolving Buffer Address Conflicts (HP 82324A Only)" later in this chapter.

Note that the statement:

```
DEVICE=HPBLP.SYS
```

provides all of the default parameters, and is equivalent to the following complete syntax:

```
DEVICE=HPBLP.SYS /1:7:250:C800 /2:9:280:C800 /3:5:330:C800
```

where:

- "/1:7:250:C800" specifies that Card 1 has interrupt level 7, I/O address 250h, and buffer address C8000h.
- "/2:9:280:C800" specifies that Card 2 has interrupt level 9, I/O address 280h, and buffer address C8000h.
- "/3:5:330:C800" specifies that Card 3 has interrupt level 5, I/O address 330h, and buffer address C8000h.

In general, these default values will work just fine. However, if you encounter a conflict, you may have to make some changes. The rest of this chapter tells you how to do this.

Resolving Interrupt and I/O Address Conflicts

The simplest type of conflict is an interrupt or I/O address conflict with another PC card. For example, suppose you have just one HP 82300C Measurement Coprocessor configured as Card 1 (IRQ7 and address 250h). But suppose you also have another card in your PC that uses IRQ7. Here are three ways you could resolve the interrupt conflict:

1. Change the interrupt setting on the other PC card. This is the simplest method as far as the measurement coprocessor is concerned, but maybe you can't or don't want to do this.
2. Change the interrupt and address hardware settings on the measurement coprocessor to match "Card 2" (IRQ 9 and I/O address 280h). In this case you won't have to change your CONFIG.SYS file.
3. Change the interrupt setting on the measurement coprocessor to another value (IRQ 9, for example), but leave the I/O address in the "Card 1" setting (250h). Then change the "DEVICE=HPBLP.SYS" statement in the CONFIG.SYS file.

If you select the third method, you can use the following syntax in your CONFIG.SYS file:

```
DEVICE=HPBLP.SYS /1:9:250:C800 /2:7:280:C800 /3:5:330:C800
```

or just:

```
DEVICE=HPBLP.SYS /1:9:250 /2:7:280 /3:5:330
```

It's all right to leave the buffer address segment ("C800") out of the statement. The default value will remain active for the HP 82324A, and the HP 82300C doesn't respond to it in any case. However, it is best to specify everything else. The reason is that it would be easy to accidentally cause a conflict. For example, the statement:

```
DEVICE=HPBLP.SYS /1:9:250
```

is really equivalent to:

```
DEVICE=HPBLP.SYS /1:9:250 /2:9:280 /3:5:330
```

In either case, both Card 1 and Card 2 are configured to IRQ 9. *This will not work, even if only one card has been installed.* By explicitly specifying the I/O addresses and interrupt levels for all three cards, whether present or not, you can ensure that no accidental conflicts are introduced.

Let's look at a second example. Suppose you have two HP 82300C Measurement Coprocessors installed as Card 1 and Card 2 (IRQ7, 250h and IRQ9, 280h), and you want to add an HP 82324A High-Performance Measurement Coprocessor as Card 3. However, suppose there is a conflict with IRQ 5 and with I/O address 330h. You could set the HP 82324A to I/O address 390h and put the following in your CONFIG.SYS file:

```
DEVICE=HPBLP.SYS /1:7:250 /2:9:280 /3:15:390
```

Card 3 is now configured for IRQ 15 and address 390h, with the default buffer address (C8000h). Note that this interrupt configuration will not work for an HP 82300C as Card 3, only an HP 82324A.

Note

Whenever you change the configuration of a "Card" in the CONFIG.SYS file, you must set the hardware configuration of the measurement coprocessor to match the software configuration. Otherwise the measurement coprocessor won't be found. For the HP 82300C Measurement Coprocessor you must set both the I/O address and the interrupt level (IRQ) configuration. For the HP 82324A High-Performance Measurement Coprocessor, only the I/O address needs to be set. Refer to the hardware installation manual that came with your measurement coprocessor for instructions on setting the hardware configuration.

Using Secondary Addressing (HP 82324A Only)

The previous section described how to change the "DEVICE=HPBLP.SYS" statement in the CONFIG.SYS file to resolve I/O address and interrupt conflicts. However, only the four primary addresses (250h, 280h, 330h, and 390h) were covered as I/O address choices. If you are using the HP 82324A High-Performance Measurement Coprocessor, you can use secondary addressing to gain additional I/O address selections.

Note

Secondary addressing provides a wide selection of additional I/O addresses for the HP 82324A High-Performance Measurement Coprocessor. However, for most applications secondary addressing is not necessary. It is recommended that you use secondary addressing only if you are not able to use a different primary address for each HP 82324A High-Performance Measurement Coprocessor.

On the HP 82324A High-Performance Measurement Coprocessor you can set the hardware configuration switch segments to select one of four primary addresses: 250h, 280h, 330h, and 390h, and one of four secondary addresses: 0, 1, 2, and 3. (Refer to *Installing the HP 82324A High-Performance Measurement Coprocessor and Accessories* for instructions.) Thus, there are 16 possible I/O addresses, as shown in the following table:

Secondary Address Setting	Primary Address Setting			
	250h	**280h**	**330h**	**390h**
0	250h	280h	330h	390h
1	650h	680h	730h	790h
2	A50h	A80h	B30h	B90h
3	E50h	E80h	F30h	F90h

If the secondary address is 0, the I/O address is simply the primary address (250h, 280h, 330h, or 390h). But, for example, if the secondary address is 2 and the primary address is 330h, the I/O address is B30h. The I/O address from the table, then, is the one to use in the "DEVICE=HPBLP.SYS" statement.

You can install up to three HP 82324A High-Performance Measurement Coprocessors in your PC, each with the same primary address, but each with a different secondary address. For example, suppose you leave each of three HP 82324A cards set to primary address 250h (the default). However, you set Card 1 to secondary address 0, Card 2 to secondary address 1, and Card 3 to secondary address 2. To use this configuration, you could put the following statement in CONFIG.SYS:

```
DEVICE=HPBLP.SYS /1:7:250:C800 /2:9:650:C800 /3:5:A50:C800
```

Note

All of your HP 82324A High-Performance Measurement Coprocessors can share the same primary address by using different secondary addresses. However, they cannot use a primary address that some other PC card uses, *including the HP 82300C Measurement Coprocessor.* The HP 82300C Measurement Coprocessor cannot respond to secondary addressing, so it must have a unique primary address.

Resolving Buffer Address Conflicts (HP 82324A Only)

The boot code for the HP 82324A High-Performance Measurement Coprocessor is loaded into RAM from PC mass storage by means of memory-mapped I/O. The boot code is transferred through a buffer within the PC memory address space. There are two possible conflicts with the buffer address:

1. *Another PC card may be set to the same buffer address.* You will need to ensure that no other PC card uses the same buffer address space as the HP 82324A High-Performance Measurement Coprocessor. However, if you have multiple HP 82324A High-Performance Measurement Coprocessor cards, all of them can use the same buffer address.
2. *An Expanded Memory Manager (EMM) or Microsoft Windows memory manager may cause a conflict with the buffer address.* If you are using an EMM or a version of Windows, you may need to "exclude" the memory address space used by the buffer.

Note

The HP 82300C Measurement Coprocessor does not use memory-mapped I/O, so neither of these conflicts will occur for it. *The following sections apply only to the HP 82324A High-Performance Measurement Coprocessor.*

Conflicts with Other Cards

The HP 82324A High-Performance Measurement Coprocessor buffer address may conflict with the buffer address of some other card that uses memory-mapped I/O. For example, there could be a conflict with the HP 82335A HP-IB Card. However, the default buffer address for the HP 82324A High-Performance Measurement Coprocessor is C8000h, and the default buffer address for the HP 82335A HP-IB Card is DC000h. Thus, no conflict will occur unless you change the buffer address of one of the cards.

Although you must make sure that the measurement coprocessor buffer address does not conflict with the buffer address of any other kind of PC card, you can use the same buffer address for all measurement coprocessors present in your system. *In fact, you should do this to avoid wasting memory address space.*

Many PC cards, like the HP 82335A HP-IB Card, use hardware switch settings to set the buffer address. However, the HP 82324A High-Performance Measurement Coprocessor buffer address is software configurable. You can specify a new buffer address for each card present, but you normally will want to specify the same buffer address for all cards. The buffer address is specified by including the *bufseg* parameter in the "DEVICE=HPBLP.SYS" statement in the CONFIG.SYS file. The *bufseg* parameter is the segment value corresponding to the buffer address, as given in the following table.

Buffer Address (hexadecimal)	**Corresponding Segment Value (*bufseg*)**
C8000 (default)	C800
CC000	CC00
D0000	D000
D4000	D400
D8000	D800
DC000	DC00

For example, to change the buffer address for each of three HP 82324A High-Performance Measurement Coprocessors to D8000h, use the following statement:

```
DEVICE=HPBLP.SYS /1:7:250:D800 /2:9:280:D800 /3:5:330:D800
```

(The statement leaves the interrupts and I/O addresses in their default values.) The new buffer address will become active when the PC is rebooted. (Save your file and then press Ctrl Alt Del to reboot.)

Note For most Vectra PCs the best buffer address is the default (C8000h). However, if you have a pre-1988 Vectra PC with the old-style HP-HIL keyboard, or if you have a Vectra 386/25 PC, you may need to change the buffer address to avoid a conflict.

Conflicts with Expanded Memory or Windows

If you are using an Expanded Memory Manager (EMM) or a version of Microsoft Windows, you may need to "exclude" the buffer address space from use by the EMM or window manager. Otherwise, an address conflict may cause your HP 82324A High-Performance Measurement Coprocessor to fail to boot. The method that you use to exclude the buffer address depends on the memory manager you are using. The rest of this section covers some common examples. If your memory manager is not listed, refer to your memory manager documentation for the correct syntax.

Note

In all cases, the exclusion will not become active until the PC is rebooted. (Save your file and then press Ctrl Alt Del to reboot.)

HPEMM386.SYS

If you are using HPEMM386.SYS, you may need to include an "EXCLUDE" option in your "DEVICE=HPEMM386.SYS" statement in the CONFIG.SYS file. The following table gives the "EXCLUDE" syntax for each buffer address:

Buffer Address (hexadecimal)	EXCLUDE Syntax
C8000 (default)	EXCLUDE=C800-CC00
CC000	EXCLUDE=CC00-D000
D0000	EXCLUDE=D000-D400
D4000	EXCLUDE=D400-D800
D8000	EXCLUDE=D800-DC00
DC000	EXCLUDE=DC00-E000

For example, if you are using the default buffer address (C8000h), put the following statement in CONFIG.SYS:

```
DEVICE=HPEMM386.SYS EXCLUDE=C800-CC00
```

HPEMMGR.SYS

If you are using HPEMMGR.SYS, you may need to include an "X" option in your "DEVICE=HPEMMGR.SYS" statement in the CONFIG.SYS file. The following table gives the "X" syntax for each buffer address:

Buffer Address (hexadecimal)	X Syntax
C8000 (default)	X=C800-CBFF
CC000	X=CC00-CFFF
D0000	X=D000-D3FF
D4000	X=D400-D7FF
D8000	X=D800-DBFF
DC000	X=DC00-DFFF

For example, if you are using the default buffer address (C8000h), put the following statement in CONFIG.SYS:

```
DEVICE=HPEMMGR.SYS X=C800-CBFF
```

Windows/386 (Version 2.03)

If you are using Windows/386 (Version 2.03 from Microsoft), you may need to include a "LASTEMMSEG" statement in the "WIN.INI" file.

Buffer Address (hexadecimal)	LASTEMMSEG Syntax
C8000 (default)	LASTEMMSEG=0C800
CC000	LASTEMMSEG=0CC00
D0000	LASTEMMSEG=0D000
D4000	LASTEMMSEG=0D400
D8000	LASTEMMSEG=0D800
DC000	LASTEMMSEG=0DC00

For example, if you are using the default buffer address (C8000h), put the following statement in WIN.INI:

```
LASTEMMSEG=0C800
```

Windows 2.11 or Windows 3.0

If you are using Windows 2.11 or Windows 3.0 from Microsoft, you may need to include an "EMMEXCLUDE" statement. This statement goes in the WIN.INI file if you are using Windows 2.11. For Windows 3.0 the statement goes in the [386Enh] section of the SYSTEM.INI file.

Buffer Address (hexadecimal)	EMMEXCLUDE Syntax
C8000 (default)	EMMEXCLUDE=0C800-0CC00
CC000	EMMEXCLUDE=0CC00-0D000
D0000	EMMEXCLUDE=0D000-0D400
D4000	EMMEXCLUDE=0D400-0D800
D8000	EMMEXCLUDE=0D800-0DC00
DC000	EMMEXCLUDE=0DC00-0E000

For example, if you are using the default buffer address (C8000h), put the following statement in WIN.INI or SYSTEM.INI:

```
EMMEXCLUDE=0C800-0CC00
```

Computer Museum

A

Using SRM with the Measurement Coprocessor

By installing an HP 50963A SRM Card in your PC, along with an HP 82300C Measurement Coprocessor or HP 82324A High-Performance Measurement Coprocessor, you can use your PC as a BASIC workstation on a Shared Resource Management (SRM) system. The SRM system is used to connect workstations to a central server.

Before you can use SRM, you must do the following:

Note The SRM Interface Card must have its node address set to a unique number given to you by the SRM system manager. The default (factory set) select code for the SRM interface card is 21. Each interface card attached to the measurement coprocessor *must* have a unique select code. Refer to the *HP 50963A SRM Coax Interface Installation Instructions*, shipped with your SRM card, for more information.

- Install the measurement coprocessor in your computer.
- Install the HP 50963A SRM Interface Card in your computer.
- Connect the SRM Interface Card to the measurement coprocessor.

You must also make sure that the DCOMM and SRM binaries are loaded into your HP BASIC system. *The measurement coprocessor INSTALL program does this automatically.*

System Concepts

Let's begin by looking at some of the concepts of the SRM system.

What Is an SRM Network?

An SRM network is a network of individual workstations connected by coaxial cable to an SRM server. The server is connected to disk drives, printers, and plotters that all workstations share. Each workstation can access data on a central database and send files to the printers and plotters. *However the workstations cannot communicate with each other.*

Shared Resource Support of the BASIC Language

You can use most BASIC statements that access local mass storage devices to access shared mass storage devices on SRM.

SRM adds two new HP BASIC mass storage commands — LOCK and UNLOCK — and adds the PROTECT option for use with the CAT statement. In addition, the PROTECT statement is used differently on SRM than with local files.

How the SRM System Manages Shared Peripheral Use

The SRM system not only provides shared access to printers and plotters, but also manages their use so that workstations never need to wait for output to be generated.

To use shared peripherals, you place files to be output into a special directory where they are held until the printer or plotter is free. The system keeps track of the order in which files arrive from the workstations, and outputs them in the same order. This method is called "spooling," and the directory where the files are kept is called the "spooler directory." Spooler directories are created for the SRM server's use when the shared peripherals are installed on the SRM system.

After a file is placed in a spooler directory, the workstation is free to do other processing. Please note, however, that the SRM system manages output spooling only. You cannot send information such as status codes or locations of the corners of paper from a plotter back to the workstation.

Booting From the SRM

If you desire, you can boot the HP BASIC language system into your measurement coprocessor from the SRM system. However, the exact procedure to use depends on the workstation power-up scheme that your SRM system manager has implemented.

Note

If you boot the HP BASIC language system into your measurement coprocessor locally (from PC mass storage), you can still access the SRM network to use shared resources such as remote mass storage and shared printers or plotters.

Automatic Configuration

The "SYSTEMS" directory on the SRM system disk contains the SRM operating system ("SYSTEM_SRM") binary files. In addition, "SYSTEMS" contains specific loading instructions for each local workstation for which an automatic boot procedure has been implemented. You can boot your measurement coprocessor from your PC mass storage or from the SRM system disk. If you want to automatically load your BASIC language system when you boot your measurement coprocessor, ask the SRM system manager to "STORE" your BASIC language system in the "SYSTEMS" directory.

System Loading from the SRM

The "SYSTEMS" directory also contains a file called "SYSTEM_LD" — the system loader file — which tells the SRM operating system to look for a file called "CONFIG_LDna". This file contains the name of the language system (HP BASIC, for example) to load into the memory of the local workstation. The suffix "na" is the node address assigned to the local workstation (which must be set on your SRM interface card). An "AUTOSTna" in the "SYSTEMS" directory contains the binaries to be loaded into the local workstation with node address "na" during an automatic boot process. If there is no "AUTOSTna", the default "AUTOST" in the root directory is used. Ask your SRM system manager to create these files for you. You will need to give him the names of the binary files you want loaded for your workstation.

Note

To ensure compatibility with the measurement coprocessor as well as the other workstations on the SRM system, replace the current "SYSTEM_LD" file with the file provided on the "Manual Examples, LIFINIT, and Selected CSUBs" disk. On the MS-DOS disk this file is named "SYSTEM.LD", but you'll have to rename it "SYSTEM_LD" when you copy it to the SRM.

If you want to over-ride the automatic boot from SRM (for example, if you want to boot locally), boot the measurement coprocessor by typing:

```
BASIC /BOOT  Enter
```

(from MS-DOS) and immediately press the spacebar. Or type:

```
BASIC /BOOT /MENU  Enter
```

In either case, all of the available system files (local and on the SRM) will be displayed. Choose the system you want by typing the two-character identifier (typically "1B", "2B", etc.) that precedes the system file name. If you do not press the space bar or use the "/MENU" option (and you do not have a "CONFIG_LDna" file), the first valid system file found by SRM is loaded.

Accessing a Shared Mass Storage Device

Your workstation accesses shared resources through the SRM server which is connected to the workstation through an SRM interface. The remote (SRM) mass storage device is identified by a remote Mass Storage Volume Specifier, or "remote MSVS" (similar to the local MSVS), which gives information about the SRM connection. The remote MSVS includes the following information:

- The device type "REMOTE", which specifies the SRM system.
- (Optional) The interface select code of your workstation's SRM interface. The default is the select code of the interface through which the boot ROM activates your workstation. (If you do not boot from the SRM, the default is the lowest select code of those available among the SRM interfaces in your workstation.)
- (Optional) The server's node address.
- (Optional) The volume name and volume password.

In general, the first step in accessing a mass storage device is to make that device the MASS STORAGE IS device. For example, execute the BASIC statement:

```
MSI ":REMOTE" (Enter)
```

SRM's Hierarchical Directory Structure

A directory is a file that is used to organize and control access to other files. The SRM operating system uses a hierarchical directory structure very similar to that used in the DFS mass storage system described in chapter 7. (One significant difference is that DFS allows either the slash, "/", or the back slash, "\", to be used in path names, but the SRM file system allows only the slash to be used.) The SRM directory structure is also very similar to the HP 9000 Series 200/300 Hierarchical File Structure (HFS).

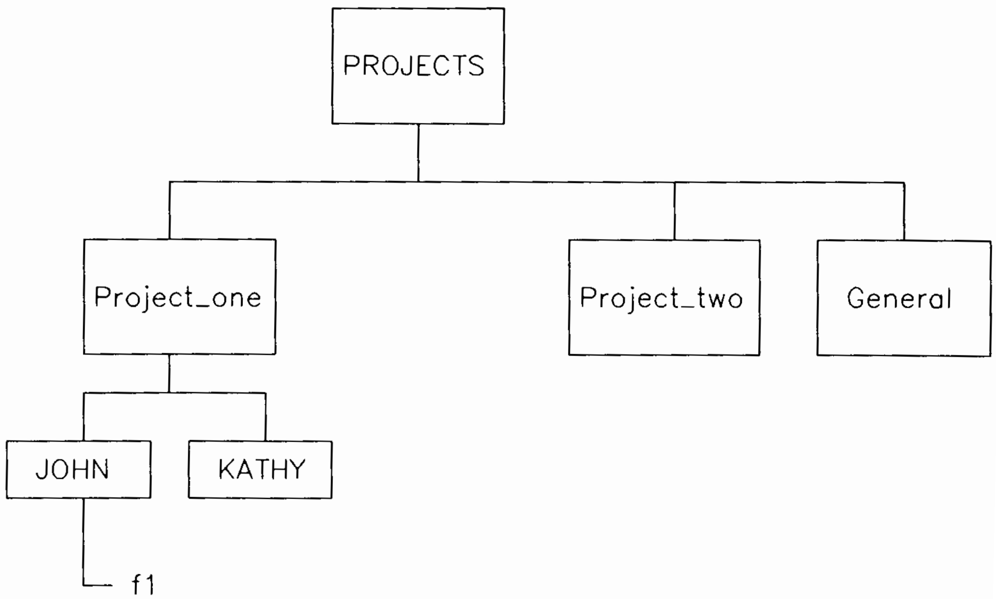
Directories are a type of file and, as such, can be:

- Created with the CREATE DIR statement. When a directory is created, its location in the hierarchical structure is fixed.
- Cataloged with the CAT statement, renamed with the RENAME statement, and protected with the PROTECT statement.
- Filled with subordinate files and directories using the COPY, CREATE, CREATE BDAT, CREATE ASCII, CREATE DIR, SAVE, STORE, RENAME, RE-SAVE, and RE-STORE statements. Each subordinate file or directory is described in its superior directory.
- Opened and closed with the MASS STORAGE IS (MSI) statement. When a user's MSI statement specifies a directory, any previously opened directory of that user is closed and the new one is opened.
- Emptied by removing all subordinate files and directories with the PURGE statement.
- Purged with the PURGE statement. You must empty and close a directory before purging it.

Using Your BASIC Workstation on SRM

This section describes, through examples, some of the more common procedures you'll use when operating your BASIC workstation on the SRM.

For the examples that follow, assume you are starting with the directory structure shown below.



Accessing the Shared Mass Storage Device

Referring to Directories and Files in the Hierarchy

To access either a directory or a file, you must specify its location in the hierarchical directory structure. This location is specified by a list of directories, called a directory path, that you must follow to reach the desired file or directory. Directory names in the list are delimited by a slash (/).

Note

Although DFS allows either the slash ("/") or the back slash ("\") to be used in path names, *the SRM file system allows only the slash ("/") to be used.*

For example, in the directory structure illustrated previously, the remote file specifier:

```
"/PROJECTS/Project_one/JOHN/f1"
```

defines the "path" to the file "f1" through its superior directories.

The path to a file begins either at the root level or at the current working directory. The working directory is the directory specified by the most recent MASS STORAGE IS statement.

Creating Directories

To create a directory named "CHARLIE" in the directory "Project_one" you could type:

```
MSI ":REMOTE" (Enter)
CREATE DIR "/PROJECTS/Project_one/CHARLIE" (Enter)
```

The leading slash indicates that the directory path begins at the root of the SRM directory structure.

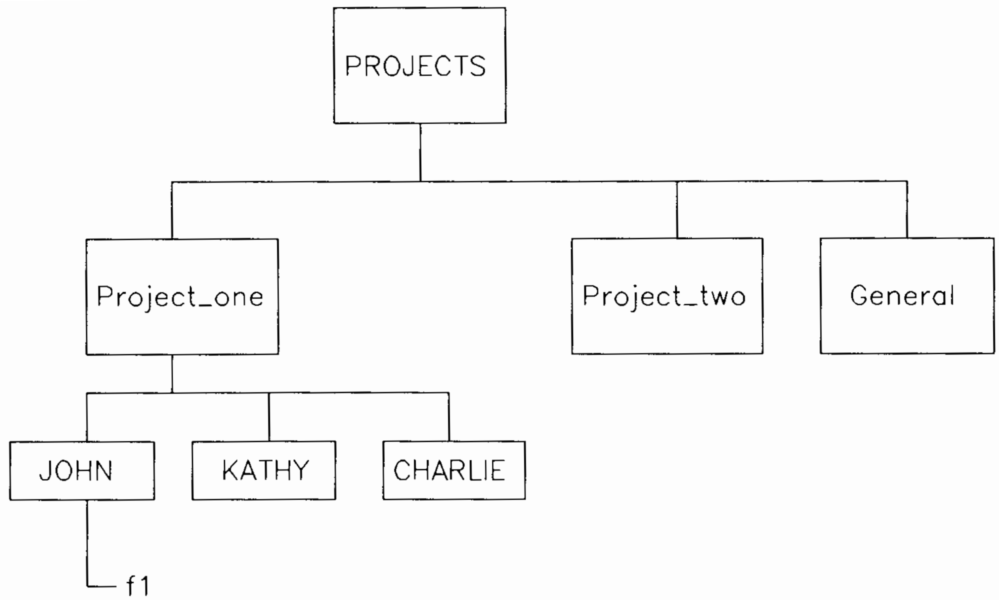
You could accomplish the same thing by typing:

```
CREATE DIR "PROJECTS/Project_one/CHARLIE:REMOTE" (Enter)
```

Using the leading slash to begin the directory path at the root works only if you have previously established the remote mass storage as the current mass storage (with some form of the

```
MSI ":REMOTE"
```

statement). This statement would place your newly-created directory into the directory structure as shown in the following illustration.



Creating Files and Other Directories Under a Directory

To create files subordinate to a new directory, you may either establish the new directory as the working directory or specify the directory path to that directory. Assuming your current working directory is the root, you could type:

```
MSI "PROJECTS/Project_one/CHARLIE"  Enter
```

to move into the directory "CHARLIE".

You could verify the new working directory with a catalog listing by typing:

```
CAT  Enter
```

The resulting listing would be as follows.

```
 PROJECTS/Project_one/CHARLIE:REMOTE 21, 0
LABEL:      Disk1
FORMAT:     SDF
AVAILABLE SPACE:       54096
                     SYS  FILE   NUMBER   RECORD      MODIFIED    PUB   OPEN
FILE NAME        LEV TYPE  TYPE  RECORDS  LENGTH DATE        TIME ACC   STAT
================ === ==== ===== ======== ======== ============== === ======
```

To create an ASCII file named "ASCII_1" that is initially to contain 100 records and be contained within "CHARLIE", you would type:

```
CREATE ASCII "ASCII_1",100 [Enter]
```

To create a BDAT file named "BDAT_1" that is initially to contain 5 records and be contained within "CHARLIE", you would type:

```
CREATE BDAT "BDAT_1",5 [Enter]
```

To create another directory called "MEMOS" within "CHARLIE", you would type:

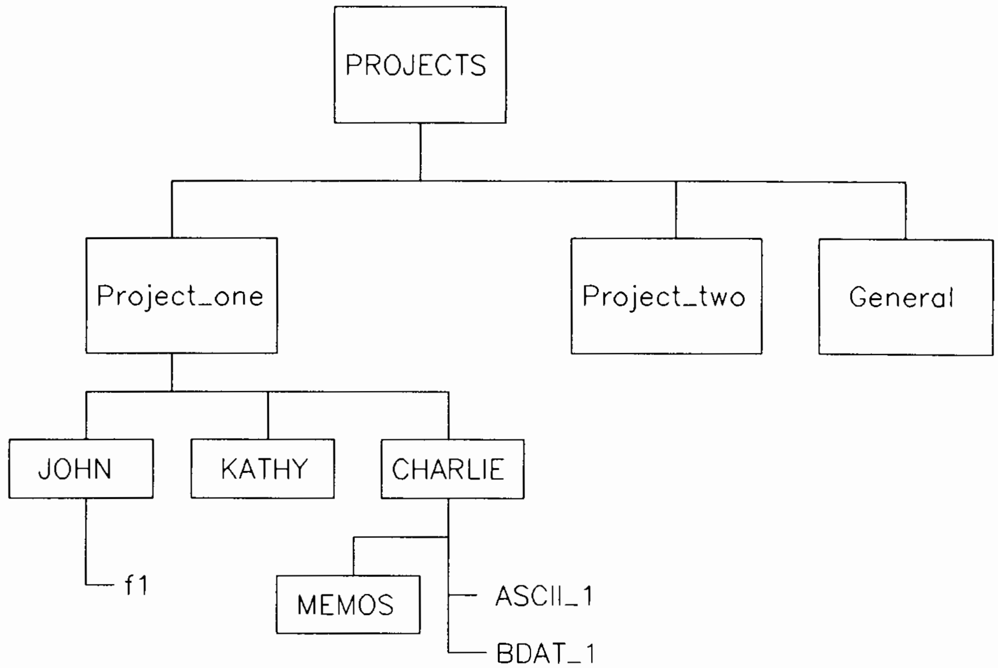
```
CREATE DIR "MEMOS" [Enter]
```

Now let's list the directory "CHARLIE" again:

```
CAT [Enter]
```

```
 PROJECTS/Project_one/CHARLIE:REMOTE 21, 0
LABEL:      Disk1
FORMAT:     SDF
AVAILABLE SPACE:       54096
                     SYS  FILE   NUMBER   RECORD      MODIFIED    PUB   OPEN
FILE NAME        LEV TYPE  TYPE  RECORDS  LENGTH DATE        TIME ACC   STAT
================ === ==== ===== ======== ======== ============== === ======
ASCII_1            1       ASCII        1      256  2-Dec-84 13:20 RWM
BDAT_1             1 98X6 BDAT         1      256  2-Dec-84 13:20 RWM
MEMOS              1       DIR          1       24  2-Dec-84 13:20 RWM
```

The additions would make the directory structure look like this:



More About the CAT Statement

The simplest form of the CAT statement:

CAT (Enter)

lists the contents of the current working directory because no directory is specifically identified. If no directory name is shown in the directory header, the current working directory is the root.

If you want to list the contents of "CHARLIE", but your current working directory is not "CHARLIE", you could:

- Designate "CHARLIE" as the working directory with the MSI statement, then use the CAT statement's "short form." For example:

  `MSI "PROJECTS/Project_one/CHARLIE:REMOTE"` (Enter)

  `CAT` (Enter)

- In the CAT statement, specify the entire path to "CHARLIE", starting at the root, by beginning the path name with a slash (/). For example:

  `CAT "/PROJECTS/Project_one/CHARLIE"` (Enter)

- This form assumes that you have already designated remote mass storage with some form of the `MSI ":REMOTE"` statement. If you have not, use the form:

  `CAT "PROJECTS/Project_one/CHARLIE:REMOTE"` (Enter)

  (The leading slash is not necessary, because including `:REMOTE` specifies the root as the beginning of the path.)

- If you were in "MEMOS" (the directory immediately subordinate to "CHARLIE",) you could use the ".." notation. For example:

  `CAT ".."` (Enter)

Shared Access to Remote Directories and Files

Because the sharing of files is a consequence of shared mass storage, the SRM system provides features for controlling access to shared information.

The SRM system offers three kinds of access capability for files and directories: READ, WRITE, and MANAGER. Capabilities are either public (available to all workstations on the SRM) or protected (available only to users who know the appropriate password).

Capabilities are protected with the PROTECT statement, which associates password(s) with one or more access capabilities. One password can be used to protect one or more capabilities. Each file or directory can have several password/capability pairs assigned to it.

Once assigned, the password protecting an access capability must be included with the file or directory specifier to execute statements requiring that access. If you don't specify the correct password when it is required, the system will report an error and deny access to the file or directory.

READ access capability for a file allows you to execute statements that read the file. READ access capability for a directory allows you to execute statements that read the file names in the directory, and to "pass through" the directory when the directory's name is included in a directory path.

For example, in the remote file specifier

```
"/PROJECTS/Project_one<READpass>/JOHN/f1"
```

including the assigned password "<READpass>" allows passage through the directory "Project_one" to allow access to its subordinate directories and files.

WRITE access capability for a file permits you to execute statements that write to the file. WRITE access capability for a directory allows you to execute statements that add to or delete from the directory's contents.

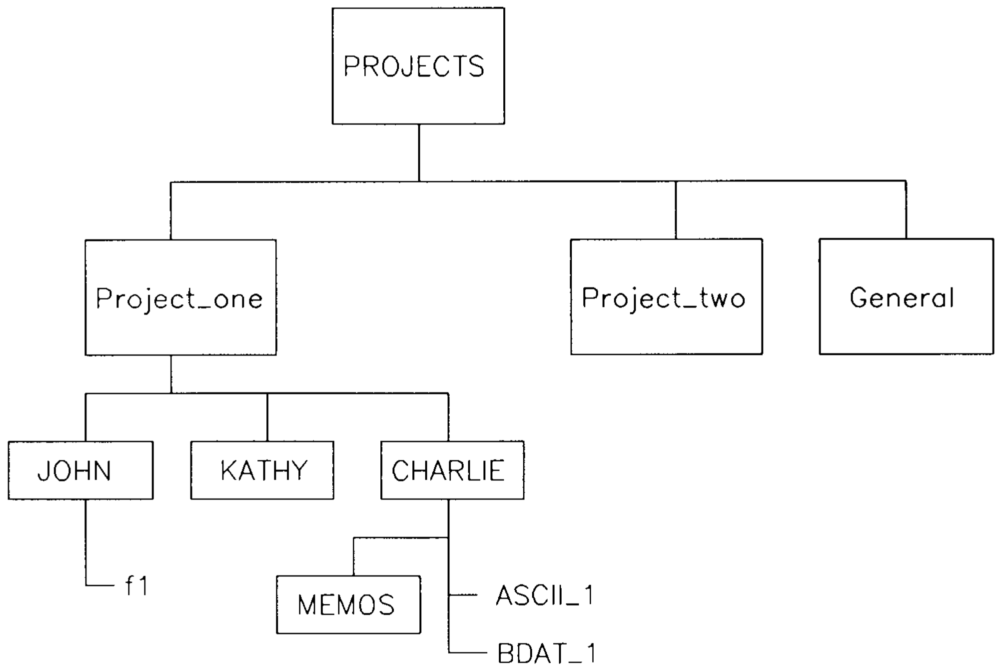
With the MANAGER access capability, public capabilities for a file or directory differ slightly from password-protected capabilities. Public MANAGER capability allows any SRM user to PROTECT, PURGE or RENAME the file. The password-protected MANAGER capability provides MANAGER, READ and WRITE access capabilities to users who include a valid password in the file or directory specifier.

Protecting Files and Directories

When you create directories and files, their access capabilities are "public" (available to any user on the SRM). You may subsequently protect a directory or file against certain types of access by other SRM workstations, provided:

- you have MANAGER access capability on the file or directory (MANAGER access to the file is public or you know the password protecting the capability),
- you have READ access capability on the directory immediately superior to the file or directory you wish to protect, and
- you protect the file or directory either while "in" its superior directory or by specifying the valid directory path to its superior directory.

For example, using the directory structure shown below, and assuming no passwords have been assigned to the files, you could:



1. Assign the password "passme" to protect the MANAGER and WRITE access capabilities on the directory "CHARLIE" with the sequence:

   ```
   MSI"/PROJECTS/Project_one" [Enter]
   PROTECT "CHARLIE",("passme":MANAGER,WRITE)  [Enter]
   ```

   which executes the PROTECT statement after moving to the directory "Project_one" (immediately superior to "CHARLIE"). As a result of the PROTECT statement, the READ access capability on "CHARLIE" is still public, but any operations that require MANAGER or WRITE capabilities must include the password.

2. Remove all public access capabilities from the file "ASCII_1" by assigning the password "no_pub", using:

```
PROTECT "CHARLIE/ASCII_1",("no_pub":MANAGER,WRITE,READ)   Enter
```

or

```
MSI "CHARLIE"   Enter
PROTECT "ASCII_1",("no_pub";MANAGER,WRITE,READ)   Enter
```

(These statements assume you are in the directory "Project_one" as if you had executed the statements in the previous step.)

The second sequence of statements makes "CHARLIE" the new working directory, whereas in the first, you merely *pass through* "CHARLIE" to reach "ASCII_1". With the READ access capability on "CHARLIE" still public, you do not need a password.

3. Protect the file "BDAT_1" so that data can be read from it, but not written into it, without using the password "write". If the current working directory were "CHARLIE", you would type:

```
PROTECT "BDAT_1",("write":MANAGER,WRITE)   Enter
```

4. Protect the MANAGER access capability of the directory "MEMOS" with the password, "mgr_pass" (so that everyone can read from and write to the directory, but a password is required to purge the directory or its contents) by typing:

```
PROTECT "MEMOS",("mgr_pass":MANAGER)   Enter
```

If you protected the files and directory in "CHARLIE" as described in the steps above, a catalog listing of "CHARLIE" would look like the following one.

```
 PROJECTS/Project_one/CHARLIE:REMOTE 21, 0
LABEL:      Disk1
FORMAT:     SDF
AVAILABLE SPACE:       54096
                          SYS  FILE   NUMBER   RECORD      MODIFIED   PUB   OPEN
FILE NAME          LEV TYPE  TYPE  RECORDS   LENGTH DATE        TIME ACC   STAT
================ === ==== ===== ======== ======== ============== === ======
ASCII_1             1       ASCII        1       256  2-Dec-84 13:20
BDAT_1              1 98X6 BDAT         1       256  2-Dec-84 13:20 R
MEMOS               1       DIR          1        24  2-Dec-84 13:20 RW
```

The letters in the column labeled "PUB ACC" indicate access capabilities that are public (not protected with a password). For example, only the MANAGER ("M") access capability on the directory "MEMOS" has been protected, leaving the READ ("R") and WRITE ("W") capabilities available to any SRM workstation user.

Specifying Passwords

When a password is required, you must include the correct password as part of the file or directory specifier in any command or statement that requires the protected access on the file or directory. The password must be enclosed between "<" and ">", and must immediately follow the name of the file or directory it protects.

For example, to get the file "ASCII_1", you might type:

```
GET "/PROJECTS/Project_one/CHARLIE/ASCII_1<no_pub>"   Enter
```

If the password were not included in the specifier, the system would respond with an error message and refuse to get the file.

Exclusive Access: Locking Files

Although sharing files saves disk space, allowing several users access to one copy of a file introduces the danger of users trying to access the file at the same time, which can cause unpredictable results. For instance, if one user tries to read part of a file while another user is writing to it, the file's contents may be inaccurate for the read.

You can "lock" a shared file with the LOCK statement, giving you sole access to that file. The same file can be locked several times in succession. Unlocking a file requires that you cancel all locks on that file. If you use the UNLOCK statement, you must cancel each LOCK with a corresponding UNLOCK. Using ASSIGN to re-open a locked file unlocks the file and you must execute another LOCK statement to lock the file again. Closing the file via

```
ASSIGN @...TO *
```

cancels all locks on the file.

In this example, a critical operation must be performed on the file named "File_a", and you do not want other users accessing the file during that operation. The program might be as follows:

```
1000   ASSIGN @File TO "File_a:REMOTE"
1010   LOCK @File;CONDITIONAL Result_code
1020   IF Result_code THEN GOTO 1010   ! Try again
1030   !  Begin critical process
  .
  .
  .
2000   !  End critical process
2010   UNLOCK @ File
```

The numeric variable called "Result_code" is used to determine the result of the LOCK operation. If the LOCK operation is successful, the variable contains 0. If the LOCK is not successful, the variable contains the numeric error code generated by attempting to lock the file.

Passwords and Protect Codes

The PROTECT statement format for remote files is different from the format for local files. Depending on the type of mass storage is being used, you can use either of the following to decide which syntax will be used:

1. Try the non-SRM syntax with an ON ERROR statement enabled. If an error occurs, see if it indicates that the mass storage device is an SRM. An Error 1 occurs when the following statement is executed on a remote file.

   PROTECT *file specifier*, *protect code*

2. If the program uses a string to store the mass storage volume specifier, check for a non-zero value of `POS (Msvs$,"REMOTE")`. This alternative is easier to implement than alternative 1 but will not work if the program accesses the default device when Msvs$ is empty.

If the program looks for a password error (Error 62) at ASSIGN time, the program may have to be modified because the system may not detect the password error until an `ENTER @Path` or `OUTPUT @Path` is attempted.

Copying Files

With SRM, you can copy files between local and remote mass storage devices by any of the methods illustrated in the following examples. Again using the directory structure established for the other examples in this section, assume that the current working directory is "CHARLIE".

Using the COPY Statement

The most direct method of copying a file from local to remote mass storage is to use the COPY statement. For example, if the current mass storage is the REMOTE directory "CHARLIE", you can copy a PROG file named "Test_prog" from local disk drive ":DOS,A" into "CHARLIE" with the statement:

```
COPY "Test_prog:DOS,A" TO "Test_prog"  (Enter)
```

By including the MSVS of the source disk (":DOS,A"), you can access the local mass storage without changing the current working directory.

The COPY statement can be used to copy files not only from local to remote mass storage, but also from remote to local mass storage and from one remote mass storage device to another. *However, you cannot copy an entire remote mass storage volume in a single COPY statement.* (You must copy a remote volume file by file.)

Suppose you want to copy the file "BDAT_1" from the directory "CHARLIE" into the directory "Project_two". Assuming the working directory is "CHARLIE", you could type:

```
COPY "BDAT_1" TO "/PROJECTS/Project_two/BDAT_1"  (Enter)
```

Using LOAD and STORE

You may also copy PROG files by loading the program into your workstation from local mass storage and then storing it in remote mass storage. For example, LOAD the program "Test_prog" from "DOS,A" into computer memory by typing:

```
LOAD "Test_prog:DOS,A"  Enter
```

Once the file is in your workstation's memory, you may then store the file in the remote directory by using a statement such as:

```
STORE "Test_prog:REMOTE"  Enter
```

Copying Item-by-Item Using ENTER and OUTPUT

You may also copy a file from local to remote mass storage an item at a time, as illustrated in the programs that follow. These programs use the ENTER and OUTPUT statements to copy data item-by-item from a local BDAT file to remote mass storage.

The first program creates and fills a BDAT file named "BDAT_FILE".

```
10  CREATE BDAT "BDAT_FILE:,1500,0",10
20  ASSIGN @Local TO "BDAT_FILE:,1500,0"
30  !
40  FOR Item=1 TO 50
50  OUTPUT @Local;"String data item"
60  NEXT Item
70  !
80  ASSIGN @Local TO *
90  END
```

The second program copies the contents of "BDAT_FILE" item-by-item into a file (also called BDAT_FILE) in the SRM directory named "General".

```
100  DIM String_item$[20]
110  CREATE BDAT "PROJECTS/General/BDAT_FILE:REMOTE",10
120  ASSIGN @Local TO "BDAT_FILE:,1500,0"
130  ASSIGN @Remote TO "PROJECTS/General/BDAT_FILE:REMOTE"
140  !
150  FOR Item=1 TO 50
160  ENTER @Local;String_item$
170  OUTPUT @Remote;String_item$'
180  NEXT Item
190  !
200  ASSIGN @Local TO *
210  ASSIGN @Remote TO *
220  END
```

Purging Remote Files and Directories

The PURGE statement works the same for removing remote files as for removing files from local mass storage. You may also remove directories using PURGE. PURGE works only with closed files and directories. Directories must also be empty (not contain any files or directories). Refer to the discussion on "Returning to Local Mass Storage" later in this section for details on closing files and directories.

When specifying the remote file to be purged, you must include all passwords protecting access capabilities required for the PURGE. For example, to purge the file "BDAT_1" from the directory "CHARLIE", you could type:

```
PURGE ".<passme>/BDAT_1<write>"  Enter
```

In this example, "CHARLIE" is the current working directory, as denoted in the directory path by the period (".").

To purge a file, you must have the MANAGER access capability on that file and READ and WRITE access capabilities on the file's superior directory. Because "passme" protects the WRITE capability on "CHARLIE" and "write" protects the MANAGER capability on "BDAT_1", both passwords must be included in the file specifier in the PURGE statement.

Although you do not normally need to specify the working directory in a directory path, you must include the password for the PURGE operation. The READ capability on "CHARLIE" is not password-protected.

To purge "CHARLIE", you would first need to purge the remaining files and directory in "CHARLIE". Because the MSI statement *opens* a directory (making it the current working directory), you must also *close* "CHARLIE".

For example, if no files or directories remained in "CHARLIE", you could purge "CHARLIE" by typing.

```
MSI ":REMOTE"  Enter
PURGE "PROJECTS/Project_one/CHARLIE<passme>  Enter
```

The first statement closes "CHARLIE" and establishes the root directory as the current working directory. Note that, because "passme" protects the MANAGER access capability on "CHARLIE", you must include that password in the PURGE statement.

Using a Shared Printer or Plotter

Use of special SRM directories called "spooler directories" allows you to access a shared printer or plotter. Setting up a spooler directory is explained in the Shared Resource Management *System Manager's Guide*. The examples in this section assume that the spooler directories "LP" (for "Line Printer") and "PL" (for "PLotter") have been created at the root of the SRM directory structure.

Spooling Using PRINTER IS and PLOTTER IS

You can use the PRINTER IS and PLOTTER IS statements to send data to your shared printer or plotter. The following command sequence illustrates this spooling method:

```
CREATE BDAT "/LP/Print_file",1
PRINTER IS "/LP/Print_file"
LIST
XREF
PRINTER IS CRT
```

The PRINTER IS and PLOTTER IS statements work only with BDAT files.

Note The DUMP DEVICE IS and PRINTALL IS statements do not support files, so they cannot be used for printer spooling.

Writing Files to the Spooler Directories

You may also access the printer associated with "LP" by placing the data to be printed in an ASCII or BDAT file in that spooler directory. For example, to list a program currently in memory, you could SAVE the program in "LP" as the file "P1_LISTING" by typing either:

```
SAVE "LP/P1_LISTING:REMOTE"  (Enter)
```

or

```
SAVE "/LP/P1_LISTING"  (Enter)
```

The SAVE statement creates an ASCII file. Although this is the same syntax used to save programs on a shared disk, the SRM system knows that "LP" is a spooler directory and prints the file as soon as possible.

Note When used for spooling, SAVE places a file in the spooler directory. The file is printed, then purged. You may wish to save or create the file first, then use the COPY statement to place the file into the spooler directory.

Sending Program Output to a Shared Printer

To spool program output to a shared printer, create an ASCII or BDAT file, assign an I/O path name to the file (which opens the file), and OUTPUT the data to that file. With BDAT files, you should ASSIGN with FORMAT ON. When the file's contents are to be printed, close the file. The following example program segment outputs the data stored in the string array called "Data$" to an ASCII file named "PERFORMANCE".

```
760  CREATE ASCII "/LP/PERFORMANCE",100
770  ASSIGN @Spool TO "/LP/PERFORMANCE"
780  OUTPUT @Spool;"Performance Summary"
790  OUTPUT @Spool;Data$(*)
800  ASSIGN @Spool TO *    ! Initiate printing.
```

The system waits until the file is not empty and closed before sending its contents to the output device. If your file is not printed or plotted within a reasonable amount of time, you may not have closed it. You can verify that your file is ready to be printed or plotted by cataloging the spooler directory:

```
CAT "/LP"  Enter
```

The open status (OPEN STAT) of the file currently being printed or plotted is listed as locked (LOCK). Files currently being written to the spooler directory (either printer or plotter) are listed as OPEN. Files that do not have a status word in the catalog are ready for printing or plotting.

The SRM 2.0 and newer operating systems allow BDAT files to be sent to the printing device as a byte stream.

Note

With the SRM 2.0 and newer operating systems, a BDAT file sent to the spooler is printed exactly as the byte stream sent. Unless you set up the BDAT file correctly, improper printer output or operation could result. Therefore, you should ASSIGN BDAT files with FORMAT ON before outputting data.

The spooler inserts a carriage return and line feed after each record in an ASCII file. To put several strings on one line, concatenate them into one string before using OUTPUT to send them to the spooler file. You may insert ASCII control characters in the data by using the CHR$ string function.

Appearance of Output

Printed output for each file includes a one-page header, which identifies the directory path to the file, the file's name, and the date and time of the printing. You can disable this header by using the "NOBANNER" option at the server.

To cause the printer to skip the paper perforation after printing a page (60 lines), prefix your file name with "FF". For example:

```
SAVE "/LP/FF_MYTEXT"  Enter
```

Aborting Printing/Plotting in Progress

To abort an in-progress printing or plotting operation, use the SPOOLER ABORT command from the SRM server. The system stops sending data to the output device and closes and purges the file. For details on bringing the spooler UP and DOWN, see the description of the SPOOLER command in the "Language Reference" section of the Shared Resource Management *System Manager's Guide*.

With SRM 2.0 and newer operating systems, if a printer is taken off-line while a file is being printed, the printer stops and resumes when the printer is put back on-line. No data is lost during such an interruption. The SRM 1.0 operating system also resumes printing, but from the beginning of the file.

Returning to Local Mass Storage

When you have finished accessing shared resources, you should close all of your files and directories to "release" the system resources.

Remote files are closed by `ASSIGN ... TO (*)`. The SCRATCH A command closes directories and files. All remote files except those opened with the PRINTER IS statement are also closed by pressing RESET.

To close your current working directory, execute an MSI to a local MSVS. For example,

```
MSI ":DOS,C"  [Enter]
```

or

```
MSI ":,1500,2"  [Enter]
```

If you booted from local mass storage, you may also execute the SCRATCH A command to completely release your access to the system. If you booted from the SRM, executing SCRATCH A resets the current working directory to the root.

Modifying Existing Programs to Access Shared Resources

If you have existing programs that are written to access local resources, you may need to modify these programs to access shared resources. This section gives some suggestions and examples that may help you.

When modifying programs to access SRM mass storage device(s), you should be aware that:

- Local and remote mass storage file specifiers may differ, and string variable names that contain file specifiers may need corresponding modification.
- References to mass storage volume specifiers throughout the program may have to be altered.
- Allowances may have to be made for directory path specification.
- Local protect codes may differ from passwords on remote files. The syntax for protecting remote files is different from that used for local files.

File Specifiers

File Name Compatibility

File names for local mass storage volumes depend on the file system:

- DFS (DOS File System) file names consist of up to eight alphanumeric characters followed by an optional period (".") and an optional "dot-extension" of up to three alphanumeric characters. All characters are "case-folded" into uppercase characters.
- LIF file names consist of up to ten alphanumeric characters and are case-sensitive. That is, lowercase and uppercase characters may be used to produce different file names. The period is not allowed in a LIF file name.

Remote (SRM) file names consist of up to 16 alphanumeric characters, are case sensitive, and may contain the period (".").

If you want to use the same file names for both remote and local mass storage, follow the requirements of your local mass storage file system since they are more restrictive than the remote file name requirements.

File and Mass Storage Device Specification in String Variables

Modifying programs for use with shared resources generally requires changing the value, and often the length, of the string variables used to specify files and mass storage devices. The statements that assign the actual values to the string variable may have to be modified individually.

Mass Storage Volume Specification

Some programs use separate variables for the file name and MSVS. For example:

```
ASSIGN @Path TO Filename$&Msvs$
```

If so, both variables may have to be dimensioned to greater lengths. Allowing 34 characters for the file name variable accommodates a 16-character file name, a 16-character password, and the "<" and ">" password delimiters (for example, "ABCDEFGHIJ123456<1234567890123456>"). The remote MSVS may occupy up to 54 characters.

Other programs may use MASS STORAGE IS statements throughout the program instead of including the MSVS in each file specifier. For instance:

```
MASS STORAGE IS Left_drive$
ASSIGN @File To File_name$
```

Unless variable(s) are used to specify the MSVS and each variable is assigned a value in only one place, you may have to modify each MASS STORAGE IS statement to specify the desired remote mass storage device.

Allowing For Directory Paths

Suppose the following program needs to be modified to include a remote file's directory path.

```
100  DIM Filename$[14],Msvs$[20]
 .
 .
 .
200  Filename$="SLIDES"
210  Msvs$=":HP9895,700"
 .
 .
 .
300  ASSIGN @File TO Filename$&Msvs$
310  OUTPUT @File;Data(*)
320  ASSIGN @File TO *
 .
 .
 .
400  ASSIGN @File TO Filename$&Msvs$
410  OUTPUT @File;Data(*)
420  ASSIGN @File to *
```

In this example, it is probably easiest to add another string variable for the (optional) path name. For example:

```
100  DIM Dir_path$[160],Filename$[80],Msvs$[80]
 .
 .
 .
200  Dir_path$="FRED/DATA_FILES/"
210  Filename$="SLIDES"
220  Msvs$=":REMOTE 21,1"
 .
 .
 .
300  ASSIGN @File TO Dirpath$&Filename$&Msvs$
310  OUTPUT @File;Data(*)
320  ASSIGN @File TO *
```


If the "Dir_path$" variable is null, the statement looks exactly like it did before the modification. If the "Msvs$" variable is null, the current mass storage device is accessed. The only difference is in the allowable length of the string variables.

In Case of Difficulty

If you have problems using the SRM network from your PC workstation, verify the following items:

- The measurement coprocessor and the SRM card are properly installed and connected.
- The measurement coprocessor is functioning correctly.
- The node address on the SRM interface card has been set to a unique number assigned by the SRM system manager.
- The SRM and DCOMM binaries are loaded in your HP BASIC system.
- See if you can access the SRM system disk from your PC workstation. Use the MSI ":REMOTE" statement to set your mass storage unit specifier to the SRM system disk, and then use CAT to display the SRM root directory.
- From the SRM server, type "NODES SC", where "SC" is the select code of the server's SRM interface card. If there is only one SRM card in the server, the select code is probably 21. The server recognizes your PC workstation if the display on the server shows the node address of your PC workstation's SRM interface card.
- Enter and run the following HP BASIC program to determine the contents of the SRM status registers:

```
10  FOR I = 1 TO 12
20    IF I = 4 THEN
30      PRINT "Register 4 not implemented."
40    ELSE
50      STATUS 21,I;Number
60      PRINT "Register ",I,"equals ",Number
70    END IF
80  NEXT I
90  END
```

  Record the results. They will help Hewlett-Packard support center personnel if you cannot find the problem.

- Run the diagnostics provided for the SRM card and record the results.

The troubleshooting section in the Shared Resource Management *System Manager's Guide* contains complete instructions for diagnosing SRM network problems. Contact your SRM manager for assistance.

Summary of SRM Status Registers

Status Register 0	Card Identification: 52 if the Remote Control switch (R) is set to 0 (closed); 180 if switch is set to 1 (open).
Status Register 1	Interface Interrupts: 1=interrupts enabled; 0=interrupts disabled.
Status Register 2	Interface Busy: 1=busy; 0=not busy.
Status Register 3	Interface Firmware ID: Always 3 (the firmware ID of the interface).
Status Register 4	Not Implemented.
Status Register 5	Data Availability: 0=receiver buffer empty. 1=receiver data available but no control blocks buffered. 2=receiver control blocks available but no data buffered. 3=both control blocks and data available.
Status Register 6	Node Address of the interface: Node address of the SRM interface installed in this computer which is set to the specified select code. The range of node addresses is 0 through 63.

Status Register 7	CRC Errors: Total number of cyclic redundancy check (CRC) errors detected by the interface since power up or RESET.
Status Register 8	Buffer Overflows: Total number of times the receive buffer has overflowed since power up or RESET.
Status Register 11	Amount of available space (number of bytes) in the transmit-data buffer.
Status Register 12	Number of transmission retries performed since power up or RESET.

B

Using a PC LAN with the Measurement Coprocessor

This appendix tells how you can transfer text and data files from your measurement coprocessor system over a PC Local Area Network (LAN). Two popular LAN software packages may be used: ARPA Services/Vectra PC and HP OfficeShare.

Sending Data Files with ARPA Services

Actually, as long as you use the DFS file system and output your data to a DOS type file, there is no problem using ARPA services. The simplest method is to run an HP BASIC program that creates the DOS type file and outputs data to it, and then exit BASIC. You can then execute an "FTP" command from MS-DOS.

The syntax for invoking an FTP command from the MS-DOS command line is as follows:

`FTP -U` *User_ID* *Password* *Remote_host_name* *FTP_command*

Let's assume that your *user ID* is "myname", your *password* is "passme" and that the name of your *remote host* is "sooner". While FTP has a large command set, the following ones are especially pertinent:

`APPEND` *DOS file remote file*	(Append DOS file to remote file.)
`CD` *remote directory*	(Change directories on remote host.)
`DELETE` *remote file*	(Delete file on remote host.)
`GET` *remote file DOS file*	(Bring file from host to MS-DOS.)
`PUT` *DOS file remote file*	(Send file from MS-DOS to host.)
`RENAME` *old file new file*	(Rename file on remote host.)
`SEND` *DOS file remote file*	(Send file from MS-DOS to host.)
`TAKE` *DOS file*	(Execute file of FTP commands.)

Note

You can use FTP to send commands and files to an HP-UX remote host, as well as to an MS-DOS remote host.

Now for an example. Let's use the FTP "SEND" command to copy the file named "DATAFILE.PRN" (in the directory C:\BLP) to the file name "datafile.cpy" (in the directory /users) in the remote host (in this case, an HP-UX system). To do this you can execute the following from the MS-DOS command line:

```
FTP -U myname passme sooner SEND C:\BLP\DATAFILE.PRN /users/datafile.cpy
```

Normally, FTP will display a running commentary on the file transfer. If you want to eliminate the commentary, you can redirect it to the "null" file as follows:

```
FTP -U myname passme sooner SEND C:\BLP\DATAFILE.PRN /users/datafile.cpy >NUL
```

You can execute FTP commands from HP BASIC through the MS-DOS communications port with an "OUTPUT 19" statement (refer to chapter 4). In this case it is a good idea to suppress the commentary. For example:

```
OUTPUT 19; "FTP -U myname passme sooner SEND C:\BLP\DATAFILE.PRN /users/datafile.cpy >NUL"
```

An HP BASIC program can communicate over the LAN by executing such "OUTPUT 19" statements. As an example, let's look at the following program, which is a modification of the first example in chapter 10:

```
10    ! This program controls an HP 3457A Multimeter at address 722.
20    !
30    CLEAR 722                         ! Clear the meter
40    OUTPUT 722; "PRESET"              ! Put meter in PRESET mode:
50                                      !    DC VOLTS (autoranging)
60                                      !    ASCII data output
70                                      !    Synchronous triggering
80    !
90    MASS STORAGE IS "\BLP:DOS,C"
100   DIM A$[128]                               ! Dimensions a string.
110   PURGE "DATAFILE.PRN"                      ! Discards previous version.
120   CREATE "DATAFILE.PRN",1                   ! Creates new DOS data file.
130   ASSIGN @File TO "DATAFILE.PRN";FORMAT ON  ! Assigns I/O path to file.
140   !
150   FOR I=1 TO 50
160     WAIT 1                          ! Takes reading every second.
170     ENTER 722;A                     ! Enters reading from DVM.
180     DISP A                          ! Displays reading.
190     IF I MOD 8=0 THEN
200       A$=VAL$(A)&CHR$(13)&CHR$(10)  ! Every 8th reading followed by CR/LF.
210     ELSE
220       A$=VAL$(A)&CHR$(32)           ! Other readings followed by a space.
230     END IF
240     OUTPUT @File USING "#,K";A$     ! Outputs data point as string to file.
250   NEXT I
260   !
270   ASSIGN @File TO *                 ! Closes I/O path.
272   !
274   ! COPY FILE TO REMOTE HOST.
276   OUTPUT 19; "FTP -U myname passme sooner SEND C:\BLP\DATAFILE.PRN /users
/datafile.cpy >NUL"
278   !
280   END
```

The OUTPUT 19 statement in line 276 sends a copy of the file to the remote host.

Outputting Data Files with OfficeShare

If you are using HP OfficeShare, the procedure is even easier. OfficeShare works by incorporating remote disk volumes into the local MS-DOS system. For example, the remote file server may appear as disk drive volume “E:” on your PC. You can create and output files directly on the remote mass storage — all you have to do is specify the path to the remote file server. The following is another modification of the example from chapter 10, but this time using OfficeShare:

```
10    ! This program controls an HP 3457A Multimeter at address 722.
20    !
30    CLEAR 722                         ! Clear the meter
40    OUTPUT 722; "PRESET"              ! Put meter in PRESET mode:
50                                      !    DC VOLTS (autoranging)
60                                      !    ASCII data output
70                                      !    Synchronous triggering
80    !
90    SPECIFY PATH TO REMOTE FILE SERVER.
100   DIM A$[128]                               ! Dimensions a string.
110   PURGE "DATAFILE.PRN:DOS,E"                ! Discards previous version.
120   CREATE "DATAFILE.PRN:DOS,E",1             ! Creates new DOS data file.
130   ASSIGN @File TO "DATAFILE.PRN:DOS,E";FORMAT ON      ! Assigns I/O path.
140   !
150   FOR I=1 TO 50
160     WAIT 1                          ! Takes reading every second.
170     ENTER 722;A                     ! Enters reading from DVM.
180     DISP A                          ! Displays reading.
190     IF I MOD 8=0 THEN
200       A$=VAL$(A)&CHR$(13)&CHR$(10)  ! Every 8th reading followed by CR/LF.
210     ELSE
220       A$=VAL$(A)&CHR$(32)           ! Other readings followed by a space.
230     END IF
240     OUTPUT @File USING "#,K";A$     ! Outputs data point as string to file.
250   NEXT I
260   !
270   ASSIGN @File TO *                 ! Closes I/O path.
280   END
```

In this case, the DOS file “DATAFILE.PRN” will be created, and the data will be output to it, in the remote volume “E:”.

C

Selected Graphics CSUBs

Four CSUBs (compiled subprograms) are provided on the "Manual Examples, LIFINIT, and Selected CSUBs" disk. These CSUBs are useful for handling raster images. GDUMP_R provides a rotated raster dump capability, while GDUMP_C provides a color raster dump capability. BPLOT lets you store and load rectangular "blocks" of raster data using numeric arrays. PCBLOCK provides a convenient means of moving, copying, or clearing a block of display graphics.

Rotated Graphics Dump (GDUMP_R)

This CSUB dumps graphics raster images to a printer. It provides the same function as the DUMP GRAPHICS statement, except that it "rotates" the image 90 degrees before sending it to the printer.

The CSUB is found in the file "GDUMP_R" on your "Manual Examples, LIFINIT, and Selected CSUBs" disk. You may want to copy this file to your system directory (for example, C:\BLP) for convenience. The CSUB is called with the statement "Gdump_rotated".

Here is an example of calling the routine to dump the image from the display raster (PLOTTER IS CRT,"INTERNAL") to a printer at device selector 701.

```
400  LOADSUB ALL FROM "GDUMP_R"  ! Load the CSUB into memory.
410  !
420  Gdump_rotated(CRT,701)      ! Dump raster (select code 1)
430                              ! to HP-IB printer (select
440                              ! code 7, address 1).
```

You can delete the CSUB from memory with the statement:

```
DELSUB "Gdump_rotated"
```

PaintJet Color Graphics Dump (GDUMP_C)

You can use the CSUB "GDUMP_COLORED" to send raster dumps of color graphics to the HP PaintJet color printer. This CSUB is a color equivalent of DUMP GRAPHICS. GDUMP_COLORED is contained in the file "GDUMP_C" on your "Manual Examples, LIFINIT, and Selected CSUBs" disk. You may want to copy this file to your system directory (for example, C:\BLP).

GDUMP_COLORED is called with several parameters. The following table describes the parameters in the order in which they appear in the CALL statement.

Parameter	Opt./Req.	Description
FROM_DS	Required	Device whose frame buffer will be dumped. No default value. It does not have to be the current "PLOTTER IS" device.
TO_DS	Required	HP-IB or RS-232 address of printer. No default value.
ROTATE$	Optional	Values: "NORMAL", "ROTATE". The default value is "NORMAL" (no rotating of image on paper; the top of the screen is the top of the printed page).
RESOLUTION	Optional	Values: 180, 90 (dots per inch). This integer value specifies the dot resolution for the printer. The default value is 180. Any value greater than 90 causes the resolution to be the default value.
BACKGROUND$	Optional	Values: "ON", "OFF". Default value is "ON" (background color will be dumped). "OFF" causes background color not to be dumped by assigning PEN 0 to white. Note that this means that all PEN 0 pixels are white, even those that may be integral parts of the picture, so take care in using this option. This parameter has no effect if you are dumping monochrome graphics (using a monochrome CRT).
ALGORITHM$	Optional	Values: "DITHER", "ERRDIF". The default value is "DITHER". For "DITHER" the raster dump algorithm uses dithering within a 2 by 2 dither cell; thus, each pixel expands to four dots on the printed page. For "ERRDIF" an error diffusion algorithm is used. Only one dot per pixel is output, so pictures are 1/4 the size of those printed with "DITHER".

Calling GDUMP_COLORED

You can call GDUMP_COLORED from within a program as shown in the following example:

```
400  LOADSUB ALL FROM "GDUMP_C"
410  !
420  Gdump_colored(CRT,701,"ROTATE",90)
430  !
```

The parameters in statement 420 have the following meanings:

- CRT sets the CRT as the device to be dumped.
- 701 is the HP-IB address of the printer.
- "ROTATE" rotates the dump on the paper.
- 90 sets the resolution in dots per inch (dpi).
- BACKGROUND$ is omitted, thus the default ("ON") is used.
- ALGORITHM$ is omitted, thus the default ("DITHER") is used.

You can call the CSUB interactively (not from a program) using the CALL statement. For example (note that the CSUB letters are all upper-case):

```
CALL GDUMP_COLORED(25,702)
```

To delete the CSUB from memory, execute the statement:

```
DELSUB GDUMP_COLORED
```

Avoid a BASIC Reset when a raster dump is being performed. The printer buffer will not be cleared and the remainder of the dump may appear on the next printout. To ensure that the printer buffer is cleared, allow the dump to complete normally. If you do use Reset, cycle the power on the printer to clear the buffer before starting the next dump.

Choosing Parameters

The following table compares the two ALGORITHM$ parameters.

DITHER	ERRDIF
Produces single-pixel-wide lines very well.	Does not produce single-pixel-wide lines well.
Can miss slight differences in shading.	Detects shading differences well.
Clips a high-resolution screen image.	Will not clip a high-resolution screen image at the default printer resolution (180 dpi).

If the image has critical parts consisting of single-pixel-wide lines, choose the "DITHER" algorithm. (You may need to specify clipping limits. Refer to "Avoiding Clipping.") If color variation is important to the image, choose "ERRDIF".

The "DITHER" Algorithm

A color dump with the "DITHER" algorithm translates each pixel on the CRT into a 2 by 2 dither cell, thus printing four dots on the PaintJet printer for every pixel. This provides five color intensity levels for each of red, green, and blue (and a maximum of 125 printable colors).

Each pixel on the CRT is described by r, g, and b color values (numeric expressions in the range 0 - 1). These r, g, and b values are partitioned into the five color intensity levels as follows:

- 0.0 - 0.12
- 0.13 - 0.37
- 0.38 - 0.62
- 0.63 - 0.87
- 0.88 - 1.0

To derive different colors on the printer, the color value must differ in level for at least one component (r, g, or b). For example, two reds with r=0.95, g=0, b=0 and r=0.90, g=0, b=0 will look the same (the r components do not fall into different levels). However, a red with r=0.85, g=0, b=0 will differ because one less dot in each dither cell will be "turned on."

The "ERRDIF" Algorithm

A raster dump with the "ERRDIF" (error diffusion) algorithm is much more sensitive to shading differences than one using "DITHER". The color of a dot depends partially on the colors of the dots around it, not solely on its own absolute color value. For more information, refer to the description of the Floyd-Steinberg error distribution algorithm in "Procedural Elements for Computer Graphics" by David F. Rogers, McGraw-Hill, 1985.

Avoiding Clipping

Clipping occurs when a screen dump has more dots (in one or both dimensions) than the printer can print. At 180 dpi the PaintJet printer can print up to 1440 dots across the width of the paper, but somewhat over 1900 dots along its length. The corresponding numbers at 90 dpi are 720 dots and 950 dots, respectively. There are three methods you can use to avoid clipping: 1) Rotate the image so that the top of the screen prints along the length of the page. 2) Use the default resolution of 180 dpi. 3) Limit the screen area to be dumped by setting clipping limits with the VIEWPORT statement.

Clipping is more likely to occur if you are using the "DITHER" algorithm since "DITHER" maps four dots per pixel. For example, for a 1024 by 768 pixel monitor, the "DITHER" algorithm produces a dump of 2048 by 1536 dots. This exceeds the maximum number of dots that can be printed at 180 dpi. Use the statement:

```
VIEWPORT 0,124,0,93
```

to set the clipping limits and specify "ROTATE", 180 dpi, and "DITHER" when you call GDUMP_COLORED.

For a 640 by 480 pixel monitor (VGA) the "DITHER" algorithm produces a dump of 1280 by 960 dots. At 180 dpi the image can be printed without clipping, either rotated or not. However, at 90 dpi the number of dots in the dump exceeds the maximum number of dots that can be printed. To fill the screen, use the statement:

```
VIEWPORT 0,99,0,75
```

to set the clipping limits and specify "ROTATE", 90 dpi, and "DITHER" when you call GDUMP_COLORED.

You can experiment with VIEWPORT to find appropriate clipping limits for monitors with other resolutions.

With the "ERRDIF" algorithm, one dot is output per pixel. Thus, either a 640 by 480 or a 1024 by 768 screen will fit on the page at 180 dpi, regardless of orientation.

Raster Store and Load (BPLOT)

This CSUB lets you store and load rectangular "blocks" of raster data (using numeric arrays). It is found in the file "BPLOT" on your "Manual Examples, LIFINIT, and Selected CSUBs" disk. You may want to copy this file to your system directory (for example, C:\BLP).

Bload and Bstore

The BPLOT CSUB provides the Bstore and Bload utilities, which are similar to the GSTORE and GLOAD statements with the following differences:

- Bstore and Bload affect *only* a specified portion of the frame buffer. (GSTORE and GLOAD affect the entire buffer.)
- Bstore and Bload also allow you to specify a replacement rule (logical operation such as AND, OR, and EXCLUSIVE OR) to combine the source and destination pixels. (GSTORE and GLOAD are *always* dominant. They overwrite any existing destination pixels.)

Subroutine "Bstore" stores a rectangular area of the frame buffer in an INTEGER array. Here is an example of using the routine:

```
Bstore(Int_array(*),X_pixels,Y_pixels)
```

The INTEGER parameters "X_pixels" and "Y_pixels" specify the width and height of the rectangular raster area to be placed in the INTEGER array "Int_array" (in *pixels*, not in the current graphics unit of measure). The data is stored in the array as one byte per pixel for all frame buffers. This array variable must be of sufficient size to store the specified pixels, or an error is reported (error 16: improper dimensions).

If part of the area is outside the current clip limits, only the portion of the frame buffer within these limits will be stored in the array — the remainder of the array will remain unchanged.

Here is another example of calling the routine, this time with an optional INTEGER parameter that specifies how the pixels are placed into the array.

```
Bstore(Int_array(*),X_pixels,Y_pixels,Rplcmt_rule)
```

The optional parameter "Rplcmt_rule" specifies the replacement rule to use in combining:

- Source bits (frame buffer with "Bstore"; array with "Bload").

  with

- Destination bits (current contents of the array with "Bstore"; frame buffer with "Bload").

Parameter Value	Effect on Destination Bits
0	All bits set to 0.
1	Source AND Destination.
2	Source AND NOT Destination.
3	Source (Default).
4	NOT Source AND Destination.
5	Destination.
6	Source EXOR Destination.
7	Source OR Destination.
8	Source NOR Destination.
9	Source EXNOR Destination.
10	NOT Destination.
11	Source OR NOT Destination.
12	NOT Source.
13	NOT Source OR Destination.
14	Source NAND Destination.
15	All bits set to 1.

If no replacement rule is specified, rule 3 (the default) is used.

Here is another example that shows two more optional REAL parameters, "X_start" and "Y_start", which specify the upper left corner of the area to be stored:

```
Bstore(Int_array(*),X_pixels,Y_pixels,Rplcmt_rule,X_start,Y_start)
```

The "X_start" and "Y_start" parameters are in *current graphics unit of measure* (not in pixels). If these parameters are not specified, the current graphics position is used.

If you want the source bits to be placed into the destination without modification, specify a value of 3 for the replacement rule parameter.

Subroutine "Bload" loads a rectangular area of the frame buffer from an INTEGER array. Here is an example of calling the routine:

```
Bload(Int_array(*),X_pixels,Y_pixels,Rplcmt_rule,X_start,Y_start)
```

This statement loads a rectangular area on the frame buffer "X_pixels" wide and "Y_pixels" high with the current contents of the INTEGER array "Int_array".

The optional REAL parameters "X_start" and "Y_start" specify the upper left corner of the destination (in *current graphics units of measure, not in pixels*). If these parameters are not specified, the current graphics position is used.

The optional INTEGER parameter "Rplcmt_rule" specifies the replacement rule to use in combining the array elements with the current contents of the affected frame buffer area. If part of the area is outside the current clip limits, only the portion of the array that maps into the current clip limits will be loaded into the frame buffer. The clipped area of the frame buffer will remain unchanged.

Using Bstore and Bload with a VGA Display

VGA display systems (and the Compaq Portable III panel display) have square pixels, as described in chapter 6. Because of this, there are no aspect ratio problems if you are using such a display. The raster aspect ratio and screen aspect ratio are the same.

Nevertheless, using Bstore and Bload is complicated by the fact that the "X_start" and "Y_start" parameters are given in the current graphics unit of measure, while "X_pixels" and "Y_pixels" are given in *pixels*. There is a way, however, to simplify the situation. You can rescale your display so that the current graphics unit of measure is equivalent to a pixel. Chapter 6 introduced the "GESCAPE" statement, which returns the parameters "Ge(2)" and "Ge(3)":

Ge(2) = Width of screen in pixels - 1 (639 for VGA)

Ge(3) = Height of screen in pixels - 1 (479 for VGA)

You can use these parameters to rescale the display with the WINDOW statement:

```
WINDOW 0,Ge(2),Ge(3),0
```

This statement puts "0,0" in the upper left corner of the screen and makes the hard clip limits equal to 639 in the x direction, 479 in the y direction. Now you can use equivalent units to calculate the physical size and the pixel size of an area to be stored with Bstore. But, there is a catch — you have to add one pixel to the size of any dimension you draw in order to capture the whole rectangle with Bstore. The reason is that Bstore counts pixels starting with "pixel number zero."

The following program is an example of this technique:

```
10    ! Bstore with VGA (Pixel_ratio = 1.0)
20    LOADSUB ALL FROM "BPLOT"
30    GINIT
40    OPTION BASE 0
50    INTEGER Ge(7)
60    INTEGER Size,A(100)
70    GESCAPE CRT,3;Ge(*)
80    WINDOW 0,Ge(2),Ge(3),0
90    MOVE 0,0
100   Size=10
110   RECTANGLE Size,Size,FILL
120   Bstore(A(*),Size+1,Size+1)
130   END
```

The WINDOW statement (line 80) rescales the display as described earlier. Line 90 moves the current position to 0,0 (the upper left corner of the screen). Line 100 sets Size = 10, and line 110 draws a 10 by 10 square and does an area fill. Line 120 executes a Bstore with both "X_pixels" and "Y_pixels" equal to "Size+1".

The same basic technique can be used to reload the image with Bload.

Using Bstore and Bload with Displays having Non-square Pixels

If you are using an EGA color display, or one of the monochrome displays with non-square pixels, the same general technique applies as with a VGA display. However, there is an additional complication. You will need to correct for the "Pixel_ratio", which was described in chapter 6.

The following program is a modification of the one just given for a VGA display:

```
10    ! Bstore with non-square pixels
20    LOADSUB ALL FROM "BPLOT"
30    GINIT
40    OPTION BASE 0
50    INTEGER Ge(7)
60    INTEGER Size,A(100)
70    GESCAPE CRT,3;Ge(*)
71    OUTPUT 19;"BLPSTATUS"
72    ENTER 19;A$,B$,C,D$,Pixel_ratio
80    WINDOW 0,Ge(2),Ge(3),0
90    MOVE 0,0
100   Size=10
110   RECTANGLE Size,INT(Size*Pixel_ratio+0.5),FILL
120   Bstore(A(*),Size+1,INT((Size*Pixel_ratio)+0.5)+1)
130   END
```

Lines 71 and 72 have been added, which obtain "BLPSTATUS" information from the MS-DOS communications port (refer to chapter 4). The only status parameter we need is the last one: "Pixel_ratio".

In line 110 the size of the rectangle in the Y direction is corrected as follows:

```
Y_size = INT(Size*Pixel_ratio+0.5)
```

(Note that 0.5 is added to give correct rounding for the INT function.)

Also, in line 120 the "Y_pixels" parameter is corrected as follows:

```
Y_Pixels = INT((Size*Pixel_ratio)+0.5)+1)
```

The same basic technique can be used to reload the image with Bload.

Raster Block Move (PCBLOCK)

This CSUB provides a convenient means of moving, copying, or clearing a block of display graphics. It is found in the file "PCBLOCK" on your "Manual Examples, LIFINIT, and Selected CSUBs" disk. You may want to copy this file to your system directory. The CSUB is loaded with the statement:

```
LOADSUB ALL FROM "PCBLOCK"
```

You can call the following routines:

Blockmove(INTEGER *Source_x0,Source_y0,Source_x1,Source_y1,Dest_x0,Dest_y0***)**

Blockcopy(INTEGER *Source_x0,Source_y0,Source_x1,Source_y1,Dest_x0,Dest_y0***)**

Blockclear(INTEGER *X0,Y0,X1,Y1***)**

Where all parameters, defined below, are expressed in *pixels*.

Source_x0	Initial (left) coordinate of display block to be moved or copied.
Source_y0	Initial (top) coordinate of display block to be moved or copied.
Source_x1	Final (right) coordinate of display block to be moved or copied.
Source_y1	Final (bottom) coordinate of display block to be moved or copied.
Dest_x0	Initial (left) coordinate of destination for move or copy.

Dest_y0	Initial (top) coordinate of destination for move or copy.
X0	Initial (left) coordinate of display block to be cleared.
Y0	Initial (top) coordinate of display block to be cleared.
X1	Final (right) coordinate of display block to be cleared.
Y1	Final (bottom) coordinate of display block to be cleared.

Note that the following restrictions apply to these routines:

- The coordinates for the routines are expressed in pixels. Thus, the display must be scaled as follows:

  `WINDOW 0,639,0,479` *For VGA displays.*
  `WINDOW 0,639,0,349` *For EGA displays.*

- The Blockmove and Blockcopy instructions have a resolution of eight pixels along the horizontal axis for placing the block at the destination. The block's *horizontal* position at the destination can't be adjusted by less than eight pixels. However, the vertical position and the block size itself are continuous.

D

Language Extension Binaries

The HP BASIC core system contains the DFS (DOS File System) and PCCRTB (bit-mapped CRT driver) binaries. In addition, the following language extensions and drivers are automatically loaded during the software installation:

Name	Description
CLOCK	Clock
COMPLEX	Complex arithmetic
CS80	CS80 disk driver
DCOMM	Datacomm interface driver
DISC	Small disk driver
EDIT	List and edit
GPIO	Interface driver
HPIB	HPIB interface driver
IO	I/O
KBD	Keyboard extensions
LEX	Lexical order
MAT	Matrix statements
MS	Mass storage
PCCRTX	CRT extensions
PCERR	Error messages

Name	Description
PCGRAPH	Graphics
PCGRAPHX	Graphics extensions
PCPDEV	Program development
SERIAL	Serial interface driver
SRM	Shared Resource Management
PCTRANS	Transfer
XREF	Cross reference

All of these binaries and drivers are loaded during the normal installation process by an "AUTOST" program.

E

Error Messages

Error Messages Generated by HP BASIC

The error messages generated by HP BASIC are listed in the "Error Messages" appendix in the HP BASIC *Language Reference* manual. The measurement coprocessor adds only one HP BASIC error message:

Error 988. HP BASIC error 988 can be generated by using a configuration file ("BLP.CON") different from the one that was accessed when HP BASIC was booted. *If you change the HP BASIC configuration file, you must reboot the measurement coprocessor.* The most common cause of error 988 is a conflict caused by a second BLP.CON file in addition to the one in the BASIC system directory ("C:\BLP"). This "extra" BLP.CON file is often found in the root directory ("C:\"). To correct the problem, delete the extra file. To avoid the problem, make sure that you run CONF.EXE in the BASIC system directory so that it will store BLP.CON correctly.

Errors during the Execution of INSTALL.EXE

`Fatal system error; exiting installation process`

This error can occur when one of the MS-DOS library routines called by the installation program returns with an error. This error indicates that something is wrong with the host PC software.

`Subdirectory Error. Could not open path`

This error occurs when the response to the destination directory prompt contains an invalid drive name or a subdirectory name that doesn't exist, and cannot be created, on the specified disk.

`Cannot install over a pre-C.00.00 release`

This error occurs if you attempt to install the measurement coprocessor software (version C.00.00 or later) in the same directory as a pre-C.00.00 version (A.00 through A.02) of the software (for example in C:\HPW).

`Unknown Display type`

The display of the host PC could not be recognized as one of the supported display types.

`Display type not set`

This error occurs when the PC display is not supported by any of the display drivers provided. This error causes the installation to be aborted.

`Insufficient DOS memory to run HP BASIC`

The available memory in the PC is not enough to load and run the HP BASIC software.

`No display file found ...`

This error may occur if the display driver file EGAVGA.DLI or MONO.DLI is not on disk 1. It may also occur when the display in use is not one of those supported by these drivers. This gives the user an opportunity to provide an alternate display driver, should one be available.

`Could not find a boot disk`

No boot disk found.

`Not enough memory`

The installation program was not able to allocate the PC RAM needed for its operation, causing the installation to be aborted. A minimum of 640 KB of on-board RAM is required (EMS is ignored).

`No disk in drive`

This error occurs when a prompt for a disk is answered by pressing (Enter) without putting a disk in the correct drive. It may also occur if the lever is not closed on a 5.25-inch flexible disk drive.

`Incorrect disk`

This error is caused by inserting a disk other than the one requested by the installation program and pressing Enter.

Errors during the Execution of BASIC.EXE

`Measurement coprocessor system fault:` *string number*

This message occurs when memory to save the alpha screen during background operation (approximately 20 KB is needed) cannot be allocated. The failing MS-DOS routine and an error code are shown.

`Measurement coprocessor not installed or not functioning`

This error message may occur if there is no measurement coprocessor installed, or if it is defective. In addition, this message may occur if the wrong measurement coprocessor "card" number is passed as a parameter to the main program; for example, if "BASIC 3" is executed when only Card 1 (address 250h and IRQ7) is present.

`Boot program failed`

This message is typically displayed when the boot program (B2.EXE) aborts with an error, such as "boot file not found" or "boot terminated by user". This message may also occur if the measurement coprocessor configuration jumpers are set incorrectly.

`Unknown option`

This error may occur when the second argument for "BASIC" is unrecognized (something other than 1, 2, or 3). The unrecognized option is shown between parentheses.

`Invalid argument`

`Measurement coprocessor number <n> must be the first argument`

The measurement coprocessor "card" number to run (1, 2, or 3) has been specified, but not in the correct syntax order.

`Unable to BOOT while in background mode`

This error occurs when the "/BOOT" option is specified in the "BASIC" command line for a measurement coprocessor that is currently running in the background mode.

`Invalid country code specified by BLPLANG environment variable`

This message occurs when the country code passed in the command line is not one of the following: FR, IT, SG, GR, SP, US, FI, SW, UK, or BE. The country code is used to set the environment variable BLPLANG, which allows the measurement coprocessor software to recognize foreign-language keyboards and collating sequences.

`Country code ignored when not booting`

This message occurs if the country code option is specified at the command line when re-entering BASIC without a boot. (This has no effect on the execution of BASIC.)

`Measurement coprocessor device driver not installed`

This error occurs when "BASIC" is run and either the line "DEVICE=HPBLP.SYS" in the "CONFIG.SYS" file has not been executed or the driver "HPBLP.SYS" is missing.

`Not enough memory`

This error occurs when the main program cannot allocate enough PC RAM to ensure the correct operation of the various software modules. The message includes the values for available memory versus required memory in KB.

Errors during the Execution of the Main Program (B3.EXE)

`Could not load ACTDISP file`

This error occurs when the file "ACTDISP" could not be found in any of the directories specified by the "BLPDIR#" or "PATH" environment variables. ("ACTDISP" is created during the measurement coprocessor software installation.)

`Fatal Error: No Mass Storage subsystem`

This error may occur if you attempt to run B3.EXE directly, bypassing BASIC.EXE.

`Measurement coprocessor version wrong for software`

This error may occur when running the current version (C.00 or higher) measurement coprocessor software in early versions of the measurement coprocessor hardware (Rev. A). This message may also occur if the measurement coprocessor configuration jumpers are set incorrectly.

`Unknown configuration record type`

This error occurs when the configuration file "BLP.CON" contains records not generated by the CONF.EXE utility. Normally you can correct the problem by editing "BLP.CON" with CONF.EXE.

`ACTDISP does not match display or DLIFONT file not found`

This error occurs when the display driver (ACTDISP) doesn't support the display currently in use, or when the DLIFONT file (the HP Roman 8 soft font) cannot be found in any of the directories specified by the "BLPDIR#" or "PATH" environment variables.

Errors during the Execution of HPBLP.SYS

`Card failed self test`

This error message indicates a hardware fault with the measurement coprocessor or its memory.

`IRQ does not respond`

This message may indicate a hardware IRQ conflict with another card. Or, for the HP 82300C Measurement Coprocessor only, the message may indicate that the software IRQ specification does not match the hardware IRQ setting.

`IRQ not valid for this card`

This message indicates that the software IRQ specification is not valid for the particular card. For example, IRQ 4 is not valid for the HP 82324A High-Performance Measurement Coprocessor.

`Error in driver parameter`

This message indicates a syntax error on the "DEVICE=HPBLP.SYS" driver line in the CONFIG.SYS file. The message is followed by an explanation of the specific error.

`Error -- IRQ xx used more than once`

This message indicates that the indicated IRQ value has been specified (either explicitly or by default) for more than one card.

`Error -- address xxxh used more than once`

This message indicates that the indicated address has been specified (either explicitly or by default) for more than one card.

F

BASIC.EXE Options and the Boot Search Order

This appendix describes the complete syntax used to execute the BASIC.EXE program. It also describes the measurement coprocessor boot search order in detail. Note that the HP 82324A High-Performance Measurement Coprocessor boot code is loaded into RAM, while the HP 82300C Measurement Coprocessor boot code is contained in ROM. However, the syntax and search order are the same for both.

BASIC.EXE Syntax

The BASIC.EXE program provides several options. The syntax of the MS-DOS command that executes BASIC.EXE is shown below, with the options enclosed in brackets ([ ]):

`BASIC` [*1*|*2*|*3*] [*options*] [*fname*[:*MSVS*]]

Where:

[*1*|*2*|*3*]	Selects a measurement coprocessor to boot. The measurement coprocessor "card" number corresponds to a logical location called "Card 1", "Card 2", or "Card 3". Each location corresponds to a base address for that measurement coprocessor. (The defaults are Card 1 = 0x250h, Card 2 = 0x280h, and Card 3 = 0x330h, as defined by the driver "HPBLP.SYS". Other addresses can be configured as described in chapter 11.) If no "card" number is specified, the first (lowest numbered) measurement coprocessor found, starting with Card 1, is started.

Other command options are:

/BOOT	Forces the execution of the boot program. If PC300 boot ROMs are present, B2.EXE is run. If Series 200 Boot ROMs are present, B1.EXE is run. *This option must be last in the list if a system filename is also specified.*
/XX	The two characters in this option stand for any of the following country codes: FR, IT, SG, GR, SP, US, FI, SW, UK, or BE. This option has the effect of setting the environment variable BLPLANG and selecting a language when BASIC is booted.
/SELFTEST SHORT	Forces the hardware self test to be run using the short RAM test. A message, "Measurement coprocessor now running selftest", is displayed while the self test executes. When the test is complete, the standard boot display appears.
/SELFTEST LONG	Forces the hardware self test to be run using the long RAM test. The only difference the user sees between the long test and the short test is the time spent waiting — about twice as long as the "short" test.
/SELFTEST SKIP	Forces the hardware self test to be skipped. The boot display shows the results of the last self test run. This selftest option is the default. *Note:* the "short" self test will be run if the block of RAM used to store the last self test results has been corrupted. If this situation occurs, the message "Running measurement coprocessor self test" is displayed and the short selftest is run.

/MENU	Forces the boot display to show all bootable files and waits for the user to select one. This is exactly the same mode of operation invoked when the user presses a key before the boot screen is displayed.
/QUIET	Suppresses the boot display unless an error occurs (such as a self test failure).
fname[:MSVS]	Selects a specific file to boot. If "fname" is specified without an MSVS, then the file refers to a boot file on the current MS-DOS drive in the current directory. You can use either the BASIC specification or the MS-DOS specification here. For example, you can use either "SYSBA514:DOS,B" or "B:SYSBA514" to specify the "SYSBA514" file on drive "B:".

Boot File Search Algorithm

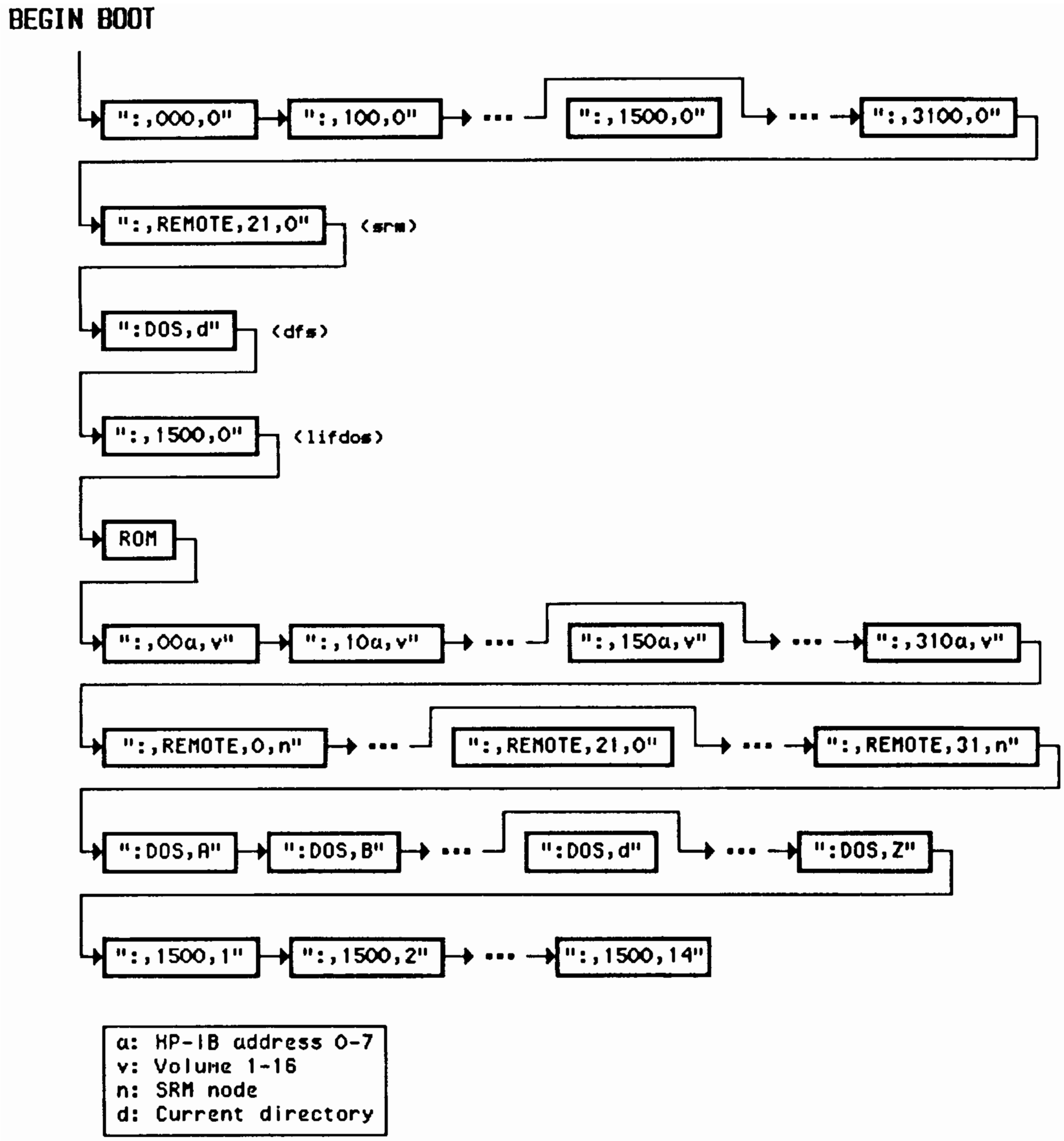
If a boot file name is not specified on the command line, the first bootable file found is booted. The PC-300 boot program, "B2.EXE", looks for bootable files using the following search algorithm:

1. External disks at select codes 0–31, bus address 0, unit 0, volume 0. Lower select codes are searched before upper select codes. *Note:* select code 15, which is emulated by the PC, is *not* searched here.
2. SRM at node 0, select code 21 on volume 8.
3. DFS boot files on the current directory on the current disk drive.
4. HPW (virtual-LIF) boot files at ":,1500,0".
5. ROM BASIC systems.

6. All remaining external disks at select codes 0–31, bus addresses 0–7, units 0–16, volumes 0–7, that were *not* searched in step 1 above.
7. All remaining SRMs at select codes 0–31 that were *not* searched in step 2 above.
8. DFS boot files in the current directories on all remaining drives, starting with A: and ending with Z:. The current drive is *not* searched here, since it was already searched in step 3 above.
9. All remaining HPW (virtual-LIF) boot files, starting at :,1500,1 and ending at :,1500,14.

This search algorithm is repeated until a boot file is selected by the user. However, once a unit is successfully scanned, it is not rescanned. For a flexible disk drive, this means that once a drive has been successfully scanned, it is not rescanned, even if a diskette is removed and another one put in its place.

The PC-300 search algorithm is illustrated on the following page.


BEGIN BOOT
":,000,0"
":,100,0"
":,1500,0"
":,3100,0"
":,REMOTE,21,0"
(srm)
":DOS,d"
(dfs)
":,1500,0"
(lifdos)
ROM
":,00a,v"
":,10a,v"
":,150a,v"
":,310a,v"
":,REMOTE,0,n"
":,REMOTE,21,0"
":,REMOTE,31,n"
":DOS,A"
":DOS,B"
":DOS,d"
":DOS,Z"
":,1500,1"
":,1500,2"
":,1500,14"
a: HP-IB address 0-7
v: Volume 1-16
n: SRM node
d: Current directory

If your measurement coprocessor has Series 200 boot ROMs, the S200 boot program, "B1.EXE", looks for bootable files using a somewhat different search algorithm:

1. External disks at select codes 0–31, bus address 0, unit 0, volume 0.
2. SRM at node 0, select code 21 on volume 8.
3. ROM BASIC systems.
4. Remaining external disks at select codes 0–31, bus addresses 0–7, units 0–16, volumes 0–7.
5. Remaining SRM systems at select codes 0–31.

Index

C

E

F

N

O

P

R

S

HEWLETT
PACKARD

HP Part Number
82324-90000

Printed in U.S.A. E1190

82324-90600 Manufacturing Number