

# Porting Guide for the Measurement Coprocessor



**HEWLETT  
PACKARD**

**HP Part No. 82324-90005  
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 ® is a U.S. registered trademark of Microsoft Corporation.

Copyright © Hewlett-Packard Company 1989, 1990, 1991

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another program language without the prior written consent of Hewlett-Packard Company.

---

## **Printing History**

First Edition — July 1989

Second Edition — November 1990

Third Edition — July 1991



# Contents

---

<b>1. The Measurement Coprocessor Software Release III</b>	
What is Release III? . . . . .	1-1
Which Release Do I Have? . . . . .	1-2
Release III Differences from Previous Releases . . . . .	1-3
Differences from Releases II and II-A . . . . .	1-3
Differences from Release I . . . . .	1-3
Differences from the Series 200/300 BASIC Workstation . . . . .	1-5
Porting Issues for Release III . . . . .	1-6
Porting from Release II or II-A . . . . .	1-6
Porting from Release I . . . . .	1-6
Porting from BASIC/WS 6.x . . . . .	1-6
<b>2. Porting from Release I to Release II-A</b>	
Mass Storage Issues . . . . .	2-1
Modifying Programs for DFS Mass Storage . . . . .	2-2
An HPW Example . . . . .	2-3
A LIF Example . . . . .	2-4
Converting Existing Files to DFS . . . . .	2-5
The CATCOPY Utility . . . . .	2-5
File Name Differences . . . . .	2-6
The COPY Statement . . . . .	2-6
Display Issues . . . . .	2-8
No More Clipping Corrections . . . . .	2-8
Combined Alpha and Graphics . . . . .	2-9
Alpha Height . . . . .	2-10
The WAIT Statement . . . . .	2-10

<b>3. Porting from Release II or II-A to Release III</b>	
Porting Issues . . . . .	3-1
HP BASIC/DOS 6.2 Enhancements . . . . .	3-2
BASIC Language Convergence . . . . .	3-3
File Related Enhancements . . . . .	3-8
Wildcards . . . . .	3-8
Appending To Files . . . . .	3-8
Overwriting Files . . . . .	3-8
A More Forgiving GET . . . . .	3-9
Human Interface Enhancements . . . . .	3-9
READ KEY . . . . .	3-9
RUNLIGHT ON/OFF . . . . .	3-9
CSUB Enhancements . . . . .	3-9
Miscellaneous Additions and Changes . . . . .	3-9
<b>4. Porting from BASIC/WS 6.x to Release III</b>	
Globalization Issues . . . . .	4-2
Mass Storage Issues . . . . .	4-2
Modifying Programs for DFS Mass Storage . . . . .	4-3
A LIF Example . . . . .	4-4
An HFS Example . . . . .	4-5
Converting Existing Files to DFS . . . . .	4-6
The CATCOPY Utility . . . . .	4-6
File Name Differences . . . . .	4-6
The COPY Statement . . . . .	4-7
Display Issues . . . . .	4-8
Alpha Height . . . . .	4-8
Series 300 Graphics Issues . . . . .	4-9
Series 200 Graphics Issues . . . . .	4-10
Other Porting Issues . . . . .	4-11
<b>A. Using the CATCOPY Utility to Convert Files</b>	
Running CATCOPY . . . . .	A-2
Choosing the “Auto” or “Manual” Copy Method . . . . .	A-5
The “Auto” Copy Method . . . . .	A-6
The “Truncate” Option . . . . .	A-6
The “Extension” Option . . . . .	A-8
The “Manual” Copy Method . . . . .	A-9

Exiting CATCOPY . . . . .	A-10
<b>B. HP BASIC Keyword Differences</b>	
Keyword Compatibility with BASIC/WS 6.2 . . . . .	B-1
Changing the Mode of the WAIT Statement . . . . .	B-10
<b>C. Porting Programs That Use CSUBs</b>	
Relinking CSUBs . . . . .	C-2
CSUB Errors . . . . .	C-2
CSUBs Provided with Release III . . . . .	C-3





# The Measurement Coprocessor Software Release III

---

Will my existing programs run? What do I need to change to make my program work the way I want it to? These are the questions you probably ask first when you start using a new version of a programming language. This chapter describes the differences between the current version of the measurement coprocessor software (Release III — HP BASIC/DOS 6.2) and the earlier versions of the software for the measurement coprocessor (formerly known as the HP BASIC Language Processor). The rest of this manual gives some *porting hints* to help you modify your programs for Release III.

*This manual does not tell you how to install or use the measurement coprocessor software. For that information, refer to *Installing and Using HP BASIC/DOS 6.2*, which is also included with the software.*

---

## What is Release III?

Release III of the measurement coprocessor software provides BASIC 6.2 functionality in the MS-DOS environment. Thus Release III is also known as “HP BASIC/DOS 6.2.” Like all previous versions of the measurement coprocessor software, Release III supports the HP 82300 Measurement Coprocessor. Like Release II-A, Release III also supports the HP 82324 High-Performance Measurement Coprocessor.

---

## Which Release Do I Have?

The following table identifies the version numbers and product numbers associated with Release I, Release II, Release II-A, and Release III of the measurement coprocessor software:

	Product Number	Software Version Number
Release I	HP 82300A	A.00.00 A.00.01
Release I	HP 82300B	A.01.00 A.02.00 A.02.01 A.02.02
Release II	HP 82300C	C.00.00 C.00.01 C.00.02
Release II-A	HP 82324A or HP 82300C	C.01.00
Release III	HP 82324B or HP 82300D	D.00.00 or higher

---

### Note



Release III allows you to install *both* HP 82300 Measurement Coprocessor and HP 82324 High-Performance Measurement Coprocessor cards in your PC at the same time. Refer to *Installing and Using HP BASIC/DOS 6.2* for further information.

---

---

## Release III Differences from Previous Releases

A great deal of effort has gone into making Release III compatible with the previous releases of the measurement coprocessor software. However, there are some differences that you should be aware of, which are summarized below.

### Differences from Releases II and II-A

Release III has changed very little from Releases II and II-A, except that BASIC 6.2 functionality has been added. The main differences are summarized below:

- BASIC 6.2 is provided, rather than BASIC 5.13 (Release II) or BASIC 5.14 (Release II-A).
- As with Release II-A, the HP 82324 High-Performance Measurement Coprocessor is supported, as well as the HP 82300 Measurement Coprocessor. (Release II did not support the HP 82324.)
- As with Release II-A, the WAIT statement has been made compatible with the Series 300 BASIC workstation. (The statement “WAIT 1” provides a delay of approximately 1 second, while for earlier releases the same statement provided a delay of approximately 0.8 second.)

### Differences from Release I

Releases II, II-A, and III *all* share the following differences from Release I:

- *Your PC and measurement coprocessor must conform to certain hardware requirements.* These requirements are listed in chapter 1 of *Installing and Using HP BASIC/DOS 6.2*.
- *BASIC in ROM is not supported.*
- *The Series 200 display model is no longer supported.* Release III does not emulate Series 200 display hardware, but provides a bit-mapped, combined alpha/graphics display like that of Series 300. Clipping is no longer a problem, but you will need to modify your Release I programs to “undo” the clipping corrections that were recommended for Release I. Refer to chapter 2 in this manual for specific instructions.
- *Blinking characters are no longer supported.*

- *The alpha display has 25 to 34 lines of alpha text, depending on the display used. (Refer to chapter 2 for specific instructions.)*
- *Each measurement coprocessor running in background requires about 100 KB of PC RAM for an MS-DOS memory resident control program. The memory resident program for background with Release I occupies only 8 KB of PC RAM, but provides much less functionality. This difference is significant if you are running a large MS-DOS application in foreground with one or more measurement coprocessors in background.*
- *The new DFS mass storage binary provides syntax differences that may require changes to your programs. You will need to change mass storage specifiers in MASS STORAGE IS statements, and in other file specifiers, to make use of DFS. Refer to chapter 2 for further information.*
- The “DGCLEAR”, “DRECTANGLE”, and “DEL\_GRAPH” CSUBs, provided with Release I, are not needed and are not supported.
- *The CONF.EXE utility has been modified. The following changes have been made. (Refer to chapter 8 in *Installing and Using HP BASIC/DOS 6.2* for detailed information.)*
  - The name of the configuration file has been changed from “HPW.CON” to “BLP.CON”.
  - The measurement coprocessor address and interrupt level are no longer configured by CONF.EXE. (This is done by the “HPBLP.SYS” driver.)
  - The “Machine Type”, “VGA Enable”, and “Background Mode” fields have been eliminated because the new display system doesn’t need them.
  - The default BLP.CON file does not configure any internal disk drives for LIF or HPW use. You will need to run CONF.EXE if you want to access LIF or HPW volumes.
  - The default *BASIC* (68000) interrupt levels for PC serial ports and PC plug-in HP-IB cards have been changed. (However, the PC interrupt levels are the same as before.)

---

## Differences from the Series 200/300 BASIC Workstation

The following comments are true for all releases of the measurement coprocessor software. They may affect you if you are porting a program from the Series 200/300 BASIC workstation (BASIC/WS) environment:

- CSUBs (compiled subprograms) that access the Series 200/300 hardware directly won't run with the measurement coprocessor. Refer to appendix C in this manual for further information.
- Serial TRANSFERs are not supported for the PC serial ports (COM1 and COM2). Also, parallel TRANSFERs are not supported for the PC parallel port (LPT1).
- Serial interrupts are not supported for COM1 and COM2.
- DMA (Direct Memory Access) is supported only by the HP 82324 High-Performance Measurement Coprocessor, not by the HP 82300 Measurement Coprocessor.
- The HP-HIL interface is supported only for HP Vectra PCs, not for non-HP computers.
- The HP 82335 and HP 82990 HP-IB interface cards are supported by the measurement coprocessor through select codes 24 and 25. HP-IB (IEEE-488) cards from other manufacturers are not supported.
- HFS file access is supported only if the HP 82313B Hierarchical File System, available separately, is installed.

---

### Note



The HP 82313A version of HFS is not compatible with HP BASIC/DOS 6.2. Only the HP 82313B version supports HP BASIC/DOS 6.2.

- 
- The following BASIC/WS binaries are *not* supported by the measurement coprocessor: “KNB2\_0”, “BCD”, “BUBBLE”, “EPROM”, “FHPIB”, “HP9885”, “LAN”, “SCSI”, “PLEL”, “MCMATH”, “CRTA”, “CRTD”, “FONT”, “INPUT”, and “LANGUAGE”.

*Refer to appendix B for a summary of keyword differences from BASIC/WS 6.2.*

---

## **Porting Issues for Release III**

The porting issues for Release III depend on the platform that you are porting from.

### **Porting from Release II or II-A**

If you have been using Release II or II-A, there are only a few porting issues that you need to consider. Most notable of these is that you will need to relink any CSUBs (compiled subprograms) that you may have created. Also, there are several enhancements to BASIC that have been introduced with BASIC 6.2. If you are porting from Release II or II-A to Release III, go directly to chapter 3.

### **Porting from Release I**

If you have been using one of the Release I versions (A.00.00 through A.02.02), there are several porting issues that you should consider. You will need to port your programs in two steps. First, follow the instructions in chapter 2 on porting from Release I to Release II-A. Then, follow the instructions in chapter 3 on porting from Release II-A to Release III.

### **Porting from BASIC/WS 6.x**

Programs written for BASIC/WS 6.x (versions 6.0, 6.1, and 6.2) should generally run without modification on the measurement coprocessor with Release III (BASIC/DOS 6.2). However, to optimize your program for the measurement coprocessor there are several porting issues that you should be aware of. These issues are covered in detail in chapter 4.

## Porting from Release I to Release II-A

---

This chapter tells how to port your programs from Release I to Release II-A of the measurement coprocessor software. This is the first step in porting to Release III (BASIC/DOS 6.2) from Release I. You don't need to have Release II-A to do this. All of the information in this chapter applies to Release III as well. *Once you have ported your Release I programs to Release II-A, go on to chapter 3 and port them to Release III.*

---

### Note



If you have written your programs for Release II or Release II-A, or if you have previously ported your programs to one of these releases, skip this chapter and go on to chapter 3.

---

There are two primary areas of interest in porting a program from Release I to Release II-A: *mass storage* and the *display*. Let's look at mass storage first.

---

### Mass Storage Issues

The measurement coprocessor Release II-A software provides three mass storage *file systems*. These are the *LIF* and *HPW* file systems, which provide compatibility with the BASIC/WS and Release I environments, and the DOS File System, or *DFS*, which provides increased access speed and direct access to the MS-DOS hierarchical file structure. *It is recommended that you use DFS file access for optimum performance and to allow direct sharing of files with MS-DOS applications.*

When you port a program from Release I, you need to do two things to convert to the DFS file system: 1) change the mass storage references and 2) copy existing files into a DFS directory. Let's look at the program modifications first.

## Modifying Programs for DFS Mass Storage

The DFS mass storage file system is covered in detail in chapter 7 of *Installing and Using HP BASIC/DOS 6.2*. In general, any statement in your Release I program that contains a MSVS (Mass Storage Volume Specifier) will need to be modified to access a DFS volume. A typical MSVS for an HPW volume would be “:,1500,0” (to access drive A:), or “:,1500,2” (to access drive C:).

---

### Note



You can continue using HPW volumes with Release II-A, but you must *enable* the disk drives that you plan to access using the utility “CONF.EXE”. (Refer to chapter 8 in *Installing and Using HP BASIC/DOS 6.2* for further information.)  
*Note, however, that the new DFS mass storage system provides significantly faster mass storage access.*

---

You can access a DFS volume with an MSVS such as “:DOS,A” (to access drive A:) or “:DOS,C” to access drive C:. However, normally you will want to subdivide your hard disk into *directories* such as “C:\BLP” and “C:\BLP\DATA”, for example. You can specify a full path in a mass storage specifier. For example, to specify the directory “C:\BLP\DATA” as the current mass storage, use the following statement.

```
MASS STORAGE IS "\BLP\DATA:DOS,C"
```

or

```
MSI "\BLP\DATA:DOS,C"
```

In addition to changing the mass storage specifiers, you may need to change the file name references in your program. DFS file names are limited to eight characters plus a “dot extension” of three characters. All alpha characters are “case-folded” into uppercase characters. Refer to “File Name Differences,” later in this chapter, for further information.



## An HPW Example

Let's port a short program segment. In the following program segment the MASS STORAGE IS statement declares “:,1500,2” (an HPW volume on drive C: in this case) to be the current mass storage:

```
340 ! Create file and store data from array "Trace_a"
350 !
360 MASS STORAGE IS ":",1500,2"
370 CREATE BDAT "DATAFILE01",13
380 !
390 ASSIGN @File TO "DATAFILE01"
400 !
410 OUTPUT @File;Trace_a(*)
420 ASSIGN @File TO *
430 !
440 END
```

This program segment creates a BDAT file named “DATAFILE01” in HPW volume “:,1500,2” and then outputs data from the array “Trace\_a” to that file.

Let's modify the program segment so that it creates a file in a DFS directory. Let's further modify the program so that it creates a data file of the *DOS* file type, rather than BDAT:

```
340 ! Create file and store data from array "Trace_a"
350 !
360 MASS STORAGE IS "\BLP\DATA:DOS,C"
370 CREATE "DATAFILE.01",1
380 !
390 ASSIGN @File TO "DATAFILE.01"
400 !
410 OUTPUT @File;Trace_a(*)
420 ASSIGN @File TO *
430 !
440 END
```

Note that line 370 creates a DOS type file with only one record. Don't worry, though. DOS files are *extensible* — they “grow” to the size needed. The file name has been modified to “DATAFILE.01”, which is a legal file name for DFS.

You can create and use BDAT files in a DFS directory. However, *the DOS file type is preferred for DFS file access* for two reasons: *higher performance* and *better compatibility* with MS-DOS applications. A DOS-type file is really a standard MS-DOS file — it has no “header.” Thus an MS-DOS application can read the file directly.

### A LIF Example

Now let's port a program written to access LIF volumes on an external HP-IB disk drive. A typical MSVS for a LIF volume on an external HP-IB disk drive would be “:,700,0” (to access drive 0 at select code 7, address 00). The following program does not specify the current mass storage device, but specifies the MSVS each time the file name is specified:

```
340  ! Create file and store data from array "Trace_a"
350  !
360  !
370  CREATE BDAT "DATAFILE:,700,0",13
380  !
390  ASSIGN @File TO "DATAFILE:,700,0"
400  !
410  OUTPUT @File;Trace_a(*)
420  ASSIGN @File TO *
430  !
440  END
```

In the following program, each LIF MSVS is replaced with the DFS MSVS “:DOS,A” to access MS-DOS disk drive “A:”. Again, line 370 is modified to create a *DOS* type file.

```
340 ! Create file and store data from array "Trace_a"
350 !
360 !
370 CREATE "DATAFILE:DOS,A",1
380 !
390 ASSIGN @File TO "DATAFILE:DOS,A"
400 !
410 OUTPUT @File;Trace_a(*)
420 ASSIGN @File TO *
430 !
440 END
```

If you want to continue using external LIF mass storage with Release II-A, you don't need to change the program at all. Also, as with Release I, no software configuration is required for external LIF mass storage. Just use the appropriate MSVS for your HP-IB disk drive.

## Converting Existing Files to DFS

If you have existing data or program files in LIF, HPW, SRM, or HFS volumes, you can convert the files to DFS by copying them to a DFS directory. You can use the HP BASIC “COPY” statement, which performs the conversion automatically. For convenience, you can also use the “CATCOPY” utility.

### The CATCOPY Utility

An HP BASIC utility program, called “CATCOPY”, is provided with Release II-A (and with Release III). You can use “CATCOPY” to copy all of the files, or those files that you choose, from a LIF or HPW volume *on an internal disk drive* to a DFS directory. The “CATCOPY” utility automatically creates a legal DFS file name for each file in the destination directory. This utility is described in appendix A of this manual.

## File Name Differences

Both the LIF and HPW file systems allow file names of up to 10 characters, with no file name extensions. LIF and HPW file names are “case-sensitive.” That is, both uppercase and lowercase characters may be used to make different file names. The following are valid LIF or HPW file names (they are all different and may exist in the same volume):

```
DATAFILE
Datafile
DATAFILE01
DATAFILE02
```

The DFS file system uses the same file name conventions as MS-DOS. That is, DFS file names may include 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. (The same rules are used in naming a DFS directory.)

Thus, all four of the above file names will map to the same DFS file name, which can cause a problem when you copy LIF or HPW files to the DFS file system. You may need to modify file names to keep each file unique. For example, the above four file names could be converted to the following for DFS:

```
DATAFILE
DATAFILE.A
DATAFILE.01
DATAFILE.02
```

SRM file names may have up to 16 characters and HFS file names may have up to 14 characters. When you copy such a file to a DFS directory, you may have to shorten the file name.

## The COPY Statement

You can use the HP BASIC “COPY” statement to copy files from one file system to another. The necessary conversions are performed automatically. For example, specify the file name and MSVS of a LIF or HPW file as the “source” and a file name and DFS directory path as the “destination”:

```
COPY "File1:,1500,0" TO "\BLP\FILE1:DOS,C" 
```

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" 
```

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" 
```

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"   
COPY "Filename01:,1500,0" TO "FILENAME.01"   
COPY "Filename02:,1500,0" TO "FILENAME.02" 
```

Here are some SRM and HFS examples:

```
COPY "/DATA/DATAFILE01:REMOTE" TO "\BLP\DATAFILE.01:DOS,C"   
COPY "/PROJECTS/DATAFILE_A:,700" TO "\BLP\DATAFILE.A:DOS,C" 
```

You can use the COPY statement to copy any file *type*: PROG, BDAT, ASCII, or HP-UX. If you copy an HP-UX file from a LIF or HPW volume (or an SRM or HFS directory), the destination file in the DFS directory will be a DOS file. Otherwise, the file type of the DFS destination file will be the same as the source file.

---

## Display Issues

The Release II-A software no longer emulates an HP 9000 Series 200 display system (for example, the 9836C) using PC hardware. Instead, a Series 300 style bit-mapped display is used, which uses the full area of the screen with no clipping of graphics images. As with Series 300, the default display mode is *combined* (merged) alpha and graphics. Let's look at some specific porting issues.

### No More Clipping Corrections

With bit-mapped graphics, you no longer need to correct for clipping of the graphics image as you did for Release I (with VIEWPORT and WINDOW statements). This means, however, that you should "undo" such corrections in any graphics programs you wrote for Release I.

If a Release I program queried the BASIC system for display specific information, it likely used statements like the following:

```
OUTPUT 19; "HPWSTATUS"  
ENTER 19; Sw_rev$,Disp_size$,Card_rev$,Machine$,Dot_ratio
```

The reason is that with Release I the standard system functions, such as SYSTEM\$, RATIO, and GESCAPE, would return the values corresponding to the *emulated hardware* (the HP 9816, HP 9836A, or HP 9836C) instead of the actual values for the PC display system.

For example, the following Release I program segment eliminates clipping and defines a proper aspect ratio for an EGA display being used for HP 9836C emulation:

```
100  ! This segment corrects for clipping for Release I.  
110  !  
120  OUTPUT 19;"HPWSTATUS"  
130  ENTER 19;Sw_rev$,Disp_size$,Card_rev$,Machine$,Dot_ratio  
140  Ega_ratio=(4/3)/(350/390)*Dot_ratio  
150  GINIT  
160  PLOTTER IS CRT,"INTERNAL"  
170  VIEWPORT 0,100*(4/3),0,100*(350/390)  
180  WINDOW 0,100*Ega_ratio,0,100
```

This correction is no longer necessary for Release II-A. You can use the normal RATIO statement just as you would with Series 300. The above Release I segment can be replaced with the following for Release II-A.

```
100  ! No clipping correction needed for Release II-A
110  !
120  GINIT
130  PLOTTER IS CRT,"INTERNAL"
140  VIEWPORT 0,100*RATIO,0,100
```

## Combined Alpha and Graphics

When you reboot the measurement coprocessor with Release II-A software, the default display mode is *Series 300 combined alpha and graphics*. The alpha and graphics planes are combined, or *merged*, into one display plane. The “ALPHA ON/OFF” and “GRAPHICS ON/OFF” statements (and the corresponding keyboard functions) have no effect. If you have a VGA or EGA display system, you can go into a Series 200 style separate alpha/graphics mode by executing the following HP BASIC statement:

```
SEPARATE ALPHA FROM GRAPHICS
```

If you execute this statement with a monochrome display (other than a VGA monochrome system), you will receive an error message.

To return to Series 300 style combined alpha and graphics, execute:

```
MERGE ALPHA WITH GRAPHICS
```

In porting your Release I programs to Release II-A, you may want to include these statements to achieve the same effect as before. Or you may want to remove the “ALPHA ON/OFF” and “GRAPHICS ON/OFF” statements from your program and use combined alpha and graphics.

*The alignment of the alpha and graphics displays will be different than that of Release I.* (This is true for both the combined and separate alpha/graphics modes.) This is because Release II-A uses the entire width of the screen for graphics, while Release I did not.

## Alpha Height

There is one other display porting issue. With Release II-A, the alpha screen has a total height (including the softkey labels) of 25 to 34 lines, depending on the type of display. For example, the VGA display has 34 lines, the EGA display has 25 lines, and the HP Multimode Display has 28 lines. To ensure that the amount of alphanumeric data displayed is not display dependent, you can include the following statement in your program:

```
ALPHA HEIGHT 25
```

---

## The WAIT Statement

For Release I (and Release II) the WAIT statement differs from HP Series 200/300 BASIC in that "WAIT 1" gives approximately an 0.8 second delay, rather than 1 second. For Release II-A, "WAIT 1" gives approximately a 1 second delay. Thus, you may need to change the WAIT statements in your programs written for Release I or II. For example, you could change "WAIT 10" to "WAIT 8" to get an 8 second delay. However, you can change the WAIT statement to be compatible with Release I or II if you want to do so, as described in appendix B.



## Porting from Release II or II-A to Release III

---

Release III of the measurement coprocessor software provides version 6.2 of HP BASIC. Thus, Release III is known as “HP BASIC/DOS 6.2”. Releases II and II-A provided versions 5.13 and 5.14, respectively, of HP BASIC. This chapter describes the additional functionality provided by HP BASIC/DOS 6.2, and identifies a few porting issues.

---

**Note**

This manual assumes that you have installed Release III in your computer following the instructions in chapter 2 of *Installing and Using HP BASIC/DOS 6.2*.

If you are porting from Release I, you will need to port to Release II-A first, as described in chapter 2.

---

---

### Porting Issues

In most cases, you can use HP BASIC/DOS 6.2 to:

- LOAD and RUN program (PROG) files created with STORE on previous versions of BASIC.
- GET and RUN program (ASCII or HP-UX) files created with SAVE on previous versions of BASIC.
- Use all data files (BDAT, ASCII, HP-UX, and DOS) created on previous versions of BASIC.

The only significant issues in porting programs from Release II-A to Release III involve compiled subprograms (CSUBs):

- If you have created CSUBs for an earlier version of BASIC, you must relink them using the HP BASIC 6.0 CSUB Utility (available separately).
- If you are using any of the CSUBs provided with the measurement coprocessor software (BPLOT, GDUMP\_C, GDUMP\_R, PCBLOCK, BLPLIB, POPLIB, ADVLIB, and 123LIB), you will have to replace them with the new versions shipped with Release III.

Refer to appendix C, “Porting Programs That Use CSUBs” for further information.

There is one additional issue in porting programs from Release II to Release III, which involves the WAIT statement. The WAIT statement for Release III (and Release II-A) works just as for workstation BASIC. That is, WAIT 1 provides approximately a 1 second wait period. However, for Release II and previous versions of the measurement coprocessor software, WAIT 1 provides approximately an 0.8 second wait period. Thus, when you port a program from Release II to Release III, you may want to adjust any WAIT statements. For example, you might change WAIT 10 to WAIT 8. (On the other hand, with Release III you can change the WAIT statement to be compatible with previous measurement coprocessor software versions if you want to do so, as described in appendix B.)

---

## HP BASIC/DOS 6.2 Enhancements

Several enhancements have been added with Release III (BASIC/DOS 6.2). The rest of this chapter describes these enhancements.

---

### Note



Because of hardware differences, BASIC/DOS 6.2 does *not* support the globalization enhancements provided by BASIC/WS 6.2 and BASIC/UX 6.2. Thus, two-byte character sets such as Japanese Kanji cannot be displayed.

---

## **BASIC Language Convergence**

A number of language extensions were added to the HP BASIC language definition with the introduction of BASIC/UX. These language extensions have also been added to BASIC/DOS 6.2 and BASIC/WS 6.2 to provide better compatibility between the three implementations of BASIC.

The operation of these keywords on BASIC/DOS systems differs in some cases from the BASIC/UX definition. Generally, BASIC/DOS operation of these keywords falls into three categories:

- Keywords that syntax and list, but do not execute.
- Keywords that syntax, list and execute, but have no effect upon system operation or return a known value.
- Keywords that syntax, list, execute and provide capability similar to the BASIC/UX definition.

This section provides a brief summary of these differences and enhancements. For a detailed description of a particular keyword, refer to the *HP BASIC 6.2 Language Reference*.

All keywords listed in the following table will syntax, LIST and LOAD on BASIC/DOS 6.2. For each BASIC keyword, the table includes information about the binary required for operation and how this keyword operates in BASIC/DOS 6.2.

### Keywords From BASIC/UX

Keyword	Binary	Execution in BASIC/DOS 6.2
CLEAR WINDOW	RMBUX	ERROR Not a in Window System
CREATE WINDOW	RMBUX	ERROR Not a in Window System
DESTROY WINDOW	RMBUX	ERROR Not a in Window System
LIST WINDOW	RMBUX	ERROR Not a in Window System
MOVE WINDOW	RMBUX	ERROR Not a in Window System
DISABLE EXT SIGNAL	RMBUX	ERROR Feature not supported in BASIC/DOS
ENABLE EXT SIGNAL	RMBUX	ERROR Feature not supported in BASIC/DOS
ON EXT SIGNAL	RMBUX	ERROR Feature not supported in BASIC/DOS
OFF EXT SIGNAL	RMBUX	ERROR Feature not supported in BASIC/DOS
ENABLE EXT SIGNAL	RMBUX	ERROR Feature not supported in BASIC/DOS
EXECUTE	RMBUX	Executes an MS-DOS command from BASIC.
QUIT or BYE	RMBUX	Exits BASIC to MS-DOS.

### Keywords From BASIC/UX (continued)

Keyword	Binary	Execution in BASIC/DOS 6.2
SET HIL MASK	None	This statement has no effect.
SET TIME	CLOCK	With no argument, this statement has no effect.
SET TIMEDATE	CLOCK	With no argument, this statement has no effect.
SYSTEM\$("PROCESS ID")	CRTX	This function returns 0.
SYSTEM\$("VERSION:OS")	None	This function returns the version of MS-DOS running on the PC.
SYSTEM\$("WINDOW SYSTEM")	None	This function returns <b>Console</b> .
TIMEZONE IS	CLOCK	With no argument, the statement has no effect.
PLOTTER IS <i>numexp</i> , "WINDOW"	GRAPH	ERROR Not in a Window System.
DUMP ALPHA <i>numexp</i> ...	None	The numeric expression must evaluate to 1 or an error will occur.
DUMP DEVICE IS <i>strex</i>	GRAPH	This is a variant of DUMP DEVICE IS where <i>strex</i> specifies a file as the destination of DUMP ALPHA or DUMP GRAPHICS. The behavior is analogous to PRINTER IS <i>file</i> .
PRINTALL IS <i>strex</i>	None	This is a variant of PRINTALL IS, where <i>strex</i> specifies a file as the printall device. The behavior is analogous to PRINTER IS <i>file</i> .

### Keywords From BASIC/UX (continued)

Keyword	Binary	Execution in BASIC/DOS 6.2
GESCAPE CRT , <i>selector</i>	GRAPHX	GESCAPE <i>selectors</i> 10-13 were added to BASIC/UX to control graphics buffering. BASIC/DOS 6.2 will accept these <i>selectors</i> , but always returns "0". These commands have no effect in BASIC/DOS.
CONTROL <i>sc</i> , 255 STATUS <i>sc</i> , 255	HPIB or GPIO	Register 255 was added to the HPIB and GPIO interfaces to control interface locking, mapping, and IO Burst extensions. BASIC/DOS accepts these registers and returns "3". These commands have no effect in BASIC/DOS.
CONTROL @ <i>path</i> , 9 STATUS @ <i>path</i> , 9	None	For paths assigned to file names, this register controls file buffering on a per file basis. ("1" = buffering enabled, "0" = buffering disabled.)
CONTROL <i>sc</i> , 22 STATUS <i>sc</i> , 22	None	ERROR Not in a Window System.
CONTROL <i>sc</i> , 23 STATUS <i>sc</i> , 23	None	This register was added to return terminal compatibility mode. BASIC/DOS accepts this register when <i>sc</i> evaluates to 1 (CRT) and always returns "1". These commands have no effect in BASIC/DOS.

### Keywords From BASIC/UX (continued)

Keyword	Binary	Execution in BASIC/DOS 6.2
<b>CONTROL 33; numexp</b> <b>STATUS 33; numexp</b>	None	This pseudo select code was added to manage external signals in BASIC/UX. BASIC/DOS does not support this select code. An error will be reported if a statement accessing select code 33 is executed.
<b>SCRATCH ALL</b> <b>SCRATCH A</b> <b>SCRATCH BIN</b> <b>SCRATCH B</b> <b>SCRATCH COM</b> <b>SCRATCH C</b> <b>SCRATCH RECALL</b> <b>SCRATCH R</b> <b>SCRATCH WINDOW</b> <b>SCRATCH W</b>	None	These statements all work the same as for BASIC/UX, except for SCRATCH WINDOW and SCRATCH W, which generate the "Not in a Window System" message.

## File Related Enhancements

### Wildcards

BASIC/DOS 6.2 supports wildcards for file name matching with many file related commands. Note that wildcard recognition is disabled at power-on and after SCRATCH A or SCRATCH BIN — you must manually enable wildcard recognition. For a detailed discussion of wildcards, refer to the WILDCARDS entry in the *HP BASIC 6.2 Language Reference*.

### Appending To Files

BASIC/DOS 6.2 allows you to specify the optional secondary keyword APPEND with many commands that write to files. When APPEND is omitted, the file is overwritten. When APPEND is specified, new data is appended to the end of the file.

The following commands support APPEND:

- ASSIGN
- DUMP DEVICE IS
- PRINTALL IS
- PRINTER IS
- PLOTTER IS

Refer to the *HP BASIC 6.2 Language Reference* for details on the proper use of APPEND with each keyword.

### Overwriting Files

BASIC/DOS 6.2 allows you to specify the secondary keyword PURGE with the following file related commands:

- COPY
- LINK (HFS only, not supported by DFS)



## **A More Forgiving GET**

BASIC/DOS 6.2 allows you to GET files that use a carriage return/line feed combination to terminate a line. This makes it easier to GET files created on certain computers, such as PCs.

## **Human Interface Enhancements**

### **READ KEY**

BASIC/DOS 6.2 provides READ KEY to allow you to read typing-aid softkey definitions created with SET KEY. Refer to the *HP BASIC 6.2 Language Reference* for a detailed description with examples.

### **RUNLIGHT ON/OFF**

BASIC/DOS 6.2 allows you to turn on or off the runlight (status indicator) at the bottom right side of the display using RUNLIGHT ON/OFF. This allows you to create more tidy displays and it is especially useful for making neat graphics dumps to printers.

## **CSUB Enhancements**

As noted before, CSUBs created for BASIC 5.0/5.1 (or earlier versions) must be relinked for BASIC/DOS 6.2. However, the internal architecture has been changed for both BASIC/WS (6.0 and later) and BASIC/DOS (6.2 and later) so that CSUBs created with the HP BASIC 6.0 CSUB Utility can be ported to later versions of BASIC *without relinking*.

## **Miscellaneous Additions and Changes**

The following are some keyword additions and changes.

<i>Keyword</i>	<i>Description</i>
INMEM	This function tests whether the specified subprogram is in memory.
CALL	The CALL statement now supports calling subprograms specified by a string, for example: CALL Subname\$ WITH (Param1,Param2\$)

## SCRATCH

The SCRATCH statement now accepts the specifiers ALL, A, BIN, B, COM, C, RECALL, and R. The specifiers WINDOW and W are also accepted, but generate the “Not in a Window System” message.

## SYSTEM\$

The SYSTEM\$ statement now allows you to request current settings for:

- WILDCARDS
- WINDOW SYSTEM
- PROCESS ID
- VERSION:OS
- VERSION:BASIC
- SYSTEM ID
- VERSION:MCP\_HW
- VERSION:MCP\_SW
- PIXEL RATIO
- DISPLAY SIZE

## Porting from BASIC/WS 6.x to Release III

---

This chapter will help you port your programs written for BASIC/WS 6.x (HP 9000 Series 200/300 workstation BASIC, version 6.0, 6.1, or 6.2) to run optimally with Release III of the measurement coprocessor software (BASIC/DOS 6.2). We say “optimally” because most programs written for BASIC/WS 6.x will run in the BASIC/DOS 6.2 environment with no changes. However, there are a few things that you may want to change to make your programs run more efficiently.

---

### Note



This manual assumes that you have installed Release III in your computer following the instructions in chapter 2 of *Installing and Using HP BASIC/DOS 6.2*. It is also assumed that you are porting programs written for BASIC/WS 6.x.

If you are porting from Series 200/300 workstation BASIC 3.x, 4.x, or 5.x you will need to port to BASIC/WS 6.x first, then to the measurement coprocessor. This may include relinking of CSUBs created for the earlier version of BASIC. Refer to *HP BASIC 6.2 Porting and Globalization* for further information.

---

There are three primary areas of interest in porting a program from BASIC/WS 6.x to BASIC/DOS 6.2: *globalization*, *mass storage*, and the *display*.

---

## Globalization Issues

BASIC/WS 6.x and BASIC/UX 6.2 support **globalization** by means of two-byte character sets (for example, Japanese Kanji). *However, BASIC/DOS 6.2 does not support globalization.* If your program makes use of a two-byte character set, you will have to modify the program to use only one-byte character sets in order for it to run in the BASIC/DOS 6.2 environment.

---

## Mass Storage Issues

The measurement coprocessor Release III software provides three mass storage *file systems*, which are described in detail in chapter 7 of *Installing and Using HP BASIC/DOS 6.2*. These are the *LIF* and *HPW* file systems, which provide compatibility with the Series 200/300 and Release I environments, and the DOS File System, or *DFS*, which provides increased access speed and direct access to the MS-DOS hierarchical file structure. *It is recommended that you use DFS file access for optimum performance and to allow direct sharing of files with MS-DOS applications.*

---

### Note



The HP 82313B Hierarchical File System is available separately for the measurement coprocessor. (Only the HP 82313B product, not HP 82313A, is compatible with Release III.) If your BASIC/WS program uses HFS, you will either have to install the HP 82313B Hierarchical File System, or you will have to modify your program to use another file system such as DFS.

---

When you write new programs, you can use DFS file access exclusively. Just follow the examples given in *Installing and Using HP BASIC/DOS 6.2*. When you port a program from BASIC/WS, you need to do two things: 1) change the mass storage references and 2) copy existing files into a DFS directory. Let's look at the program modifications first.

## Modifying Programs for DFS Mass Storage

The DFS mass storage file system is covered in detail in chapter 7 of *Installing and Using HP BASIC/DOS 6.2*. In general, any statement in your BASIC/WS program that contains a MSVS (Mass Storage Volume Specifier) will need to be modified to access a DFS volume. A typical MSVS for a LIF volume would be “:,700,0” (to access drive 0 on an external HP-IB disk drive at select code 7, address 00).

---

### Note



You can access LIF volumes on an external HP-IB disk drive with Release III. You won't need to modify your BASIC/WS programs unless the address or drive number changes. No software configuration is necessary for an external disk drive.

You can also access LIF flexible disk volumes using an internal flexible disk drive, but you must first configure the drive using the utility “CONF.EXE” as described in chapter 8 of *Installing and Using HP BASIC/DOS 6.2*. *Note, however, that the new DFS mass storage system provides significantly faster mass storage access.*

---

You can access a DFS volume with an MSVS such as “:DOS,A” (to access drive A:) or “:DOS,C” to access drive C:. However, normally you will want to subdivide your hard disk into *directories* such as “C:\BLP” and “C:\BLP\DATA”, for example. You can specify a full path in a mass storage specifier. For example, to specify the directory “C:\BLP\DATA” as the current mass storage, use the following statement.

```
MASS STORAGE IS "\BLP\DATA:DOS,C"
```

or

```
MSI "\BLP\DATA:DOS,C"
```

In addition to changing the mass storage specifiers, you may need to change the file name references in your program. DFS file names are limited to eight characters plus a “dot extension” of three characters. All alpha characters are “case-folded” into uppercase characters. Refer to “File Name Differences,” later in this chapter, for further information.

## A LIF Example

Let's port a short program segment. In the following program segment the MASS STORAGE IS statement declares “:,700,0” to be the current mass storage:

```
340 ! Create file and store data from array "Trace_a"
350 !
360 MASS STORAGE IS ":",700,0"
370 CREATE BDAT "DATAFILE01",13
380 !
390 ASSIGN @File TO "DATAFILE01"
400 !
410 OUTPUT @File;Trace_a(*)
420 ASSIGN @File TO *
430 !
440 END
```

This program segment creates a BDAT file named “DATAFILE01” in LIF volume “:,700,0” and then outputs data from the array “Trace\_a” to that file.

Let's modify the program segment so that it creates a file in a DFS directory. Let's further modify the program so that it creates a data file of the *DOS* file type, rather than BDAT:

```
340 ! Create file and store data from array "Trace_a"
350 !
360 MASS STORAGE IS "\BLP\DATA:DOS,C"
370 CREATE "DATAFILE.01",1
380 !
390 ASSIGN @File TO "DATAFILE.01"
400 !
410 OUTPUT @File;Trace_a(*)
420 ASSIGN @File TO *
430 !
440 END
```

Note that line 370 creates a DOS type file with only one record. Don't worry, though. DOS files are *extensible* — they “grow” to the size needed. The file name has been modified to “DATAFILE.01”, which is a legal file name for DFS.

You can create and use BDAT files in a DFS directory. However, the DOS file type is preferred for DFS file access for two reasons: *higher performance* and *better MS-DOS compatibility*. A DOS-type file is really a standard MS-DOS file — it has no “header.” An MS-DOS application can read the file directly provided only ASCII data has been written to the file. Refer to chapter 10 in *Installing and Using HP BASIC/DOS 6.2* for details.

### An HFS Example

You may have noticed that DFS is very similar to HFS (Hierarchical File System). Let’s port a program written for HFS file access to use DFS file access:

```
340 ! Create file and store data from array "Trace_a"
350 !
360 !
370 CREATE BDAT "/PROJECTS/DATAFILE:,700,0",13
380 !
390 ASSIGN @File TO "/PROJECTS/DATAFILE:,700,0"
400 !
410 OUTPUT @File;Trace_a(*)
420 ASSIGN @File TO *
430 !
440 END
```

In the following program, each HFS path and MSVS is replaced with one for a DFS directory. Note that HFS paths always use the forward slash (/), but DFS paths may use either the back slash (\) or the forward slash (/).

```
340 ! Create file and store data from array "Trace_a"
350 !
360 !
370 CREATE "\\BLP\\DATA\\DATAFILE:DOS,C",1
380 !
390 ASSIGN @File TO "\\BLP\\DATA\\DATAFILE:DOS,C"
400 !
410 OUTPUT @File;Trace_a(*)
420 ASSIGN @File TO *
430 !
440 END
```

## Converting Existing Files to DFS

If you have existing data or program files in LIF, SRM, or HFS volumes, you can convert the files to DFS by copying them to a DFS directory. You can use the HP BASIC “COPY” statement, which performs the conversion automatically. For convenience, you can also use the “CATCOPY” utility.

### The CATCOPY Utility

An HP BASIC utility program, called “CATCOPY”, is provided with Release III. You can use “CATCOPY” to copy all of the files, or those files that you choose, from a LIF volume *on an internal disk drive* to a DFS directory. The “CATCOPY” utility automatically creates a legal DFS file name for each file in the destination directory. This utility is described in appendix A of this manual.

### File Name Differences

The LIF file system allows file names of up to 10 characters, with no file name extensions. LIF file names are “case-sensitive.” That is, both uppercase and lowercase characters may be used to make different file names. The following are valid LIF file names (they are all different and may exist in the same volume):

DATAFILE  
Datafile  
DATAFILE01  
DATAFILE02

The DFS file system uses the same file name conventions as MS-DOS. That is, DFS file names may include 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. (The same rules are used in naming a DFS directory. Refer to chapter 7 in *Installing and Using HP BASIC/DOS 6.2* for a complete discussion of this topic.)



---

**Note**

MS-DOS *reserves* certain file names and extensions, which have a special meaning. Refer to your MS-DOS documentation for further information.

---

Thus, all four of the above file names will map to the same DFS file name, which can cause a problem when you copy LIF files to the DFS file system. You may need to modify file names to keep each file unique. For example, the above four file names could be converted to the following for DFS.

```
DATAFILE
DATAFILE.A
DATAFILE.01
DATAFILE.02
```

SRM file names may have up to 16 characters and HFS file names may have up to 14 characters. When you copy such a file to a DFS directory, you may have to shorten the file name.

### The COPY Statement

You can use the HP BASIC “COPY” statement to copy files from one file system to another. The necessary conversions are performed automatically. For example, specify the file name and MSVS of a LIF file as the “source” and a file name and DFS directory path as the “destination”. (In this example, the source is a file on an HP-IB LIF disk drive.)

```
COPY "File1:,700,0" TO "\BLP\FILE1:DOS,C" 
```

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:,700,0" TO "\BLP\Filename01:DOS,C" 
```

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:,700,0" TO "\BLP\FILENAME.01:DOS,C" 
```

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"   
COPY "Filename01: ,700,0" TO "FILENAME.01"   
COPY "Filename02: ,700,0" TO "FILENAME.02" 
```

Here are some SRM and HFS examples:

```
COPY "/DATA/DATAFILE01:REMOTE" TO "\BLP\DATAFILE.01:DOS,C"   
COPY "/PROJECTS/DATAFILE_A: ,700" TO "\BLP\DATAFILE.A:DOS,C" 
```

You can use the COPY statement to copy any file *type*: PROG, BDAT, ASCII, or HP-UX. If you copy an HP-UX file from a LIF volume (or an SRM or HFS directory), the destination file in the DFS directory will be a DOS file. Otherwise, the file type of the DFS destination file will be the same as the source file.

---

## Display Issues

When porting to Release III (BASIC/DOS 6.2), the graphics issues depend on whether you are porting from Series 300 or from Series 200. There is one display issue that is applicable to both Series 300 and Series 200, however. That is the issue of alpha height.

### Alpha Height

With Release III, the alpha screen has a total height (including the softkey labels) of 25 to 34 lines, depending on the type of display. For example, the VGA display has 34 lines, the EGA display has 25 lines, and the HP Multimode Display has 28 lines. To ensure that the amount of alphanumeric data displayed is not display dependent, you can include the following statement in your program:

```
ALPHA HEIGHT 25
```

## Series 300 Graphics Issues

Release III uses a Series 300 style bit-mapped display, which uses the full area of the screen with no clipping of graphics images. As with Series 300, the default display mode is *combined* (merged) alpha and graphics. Thus, you won't need to modify your graphics programs in most cases. You may want to refer to chapter 6 of *Installing and Using HP BASIC/DOS 6.2* for a detailed description of the bit-mapped display.

There are very few porting issues for Series 300 graphics, *provided you are using a VGA monitor*. For other monitors, there are some aspect ratio issues, which are described in chapter 6 of *Installing and Using HP BASIC/DOS 6.2*. The VGA color display provides good compatibility with the color-mapped-mode pen colors produced by Series 300 (pens 0 through 15). However, if you are using an EGA display, pen colors 8 through 15 are not a good match. *For optimum compatibility with Series 300 color graphics, a VGA color display system is highly recommended.*

If you are using a gray-scale VGA display, there are two possible mappings of the gray scale for pens 0 through 7 — default and NTSC. Chapter 6 of *Installing and Using HP BASIC/DOS 6.2* describes these two modes. Select the NTSC mode for compatibility with the HP 9000 Model 362/382 gray-scale display. The NTSC mode matches the 362/382 for pens 0 through 7, but pens 8 through 255 are not supported.

There is another issue that may affect some programs. The alignment of the alpha and graphics displays is dependent on the display hardware in use. This is true for both the measurement coprocessor and for Series 300. Thus, if you have written a program that mixes alpha text with graphic images, the positioning of the alpha and graphics elements will be different than for Series 300. You will have to experiment to achieve the desired effect. (This is true for both the combined and separate alpha/graphics modes.)

## Series 200 Graphics Issues

When you are porting a program written for a Series 200 computer to Release III, you will have to confront the same issue that arises when you port from Series 200 to Series 300: *combined* alpha and graphics planes.

When you reboot the measurement coprocessor with Release III software, the default display mode is *Series 300 combined alpha and graphics*. The alpha and graphics planes are combined, or *merged*, into one display plane. The “ALPHA ON/OFF” and “GRAPHICS ON/OFF” statements (and the corresponding keyboard functions) have no effect. If you have a VGA or EGA display system, you can go into a Series 200 style separate alpha/graphics mode by executing the following HP BASIC statement:

```
SEPARATE ALPHA FROM GRAPHICS
```

If you execute this statement with a monochrome display (other than a VGA monochrome system), you will receive an error message.

To return to Series 300 style combined alpha and graphics, execute:

```
MERGE ALPHA WITH GRAPHICS
```

In porting your Series 200 programs to Release III, you may want to include these statements to achieve the same effect as before. Or you may want to remove the “ALPHA ON/OFF” and “GRAPHICS ON/OFF” statements from your program and use Series 300 style combined alpha and graphics.

There is another issue that may affect some programs. The alignment of the alpha and graphics displays is dependent on the display hardware in use. Thus, if you have written a program that mixes alpha text with graphic images, the positioning of the alpha and graphics elements will be different than for Series 200. You will have to experiment to achieve the desired effect.

---

## Other Porting Issues

Appendix B summarizes some keyword differences for Release III compared with BASIC/WS 6.x. The table includes enhancements as well as functions not implemented. In particular, you should note the following:

- You cannot select CRT binaries with CRT Register 21. Only one CRT binary (CRTB) is supported by BASIC/DOS 6.2.
- Serial TRANSFERS are not supported for COM1 and COM2.
- Parallel TRANSFERS are not supported for LPT1.
- Serial interrupts are not supported for COM1 and COM2.
- The SOUND keyword is not supported. (Use BEEP instead.)
- The READIO and WRITEIO statements access hardware registers and memory locations directly. Thus, a detailed knowledge of the hardware of the measurement coprocessor is necessary. If you have written a BASIC/WS program using READIO and WRITEIO, it is likely that the statements won't work due to hardware differences. It is recommended that you use the STATUS and CONTROL statements instead.

---

### Caution



It is possible to damage your measurement coprocessor system by writing to certain memory locations with WRITEIO. Hewlett-Packard cannot be held liable for any damages caused by improper use of this feature.

---

## Using the CATCOPY Utility to Convert Files

---

The “CATCOPY” utility is an interactive HP BASIC program that allows you to copy files from an HPW (virtual-LIF) or LIF disk volume to a DFS directory. With CATCOPY you can copy all of the files in an HPW or LIF volume to a DFS directory in a single batch process. Or you can use the “interactive” mode and copy only those files that you choose. In either case, the *source* can be either an HPW volume or a LIF volume in one of the *internal* PC disk drives (select code 15). The *destination* can be any DFS directory that you specify.

---

### Note



You cannot use CATCOPY to copy files from an external disk drive such as an HP-IB disk drive at select code 7. If you want to copy files from a LIF flexible disk, configure one of the internal PC flexible disk drives with the “CONF.EXE” utility and use that drive as the source drive. You cannot use CATCOPY to copy files from an SRM or HFS directory. (Use the COPY statement instead.)

---

Before you run the CATCOPY program, you need to configure the internal disk drive or drives that you want to access through select code 15. Use the “CONF.EXE” utility to do this, as described in chapter 8 of *Installing and Using HP BASIC/DOS 6.2*.

If you have Release I installed in a directory such as “C:\HPW”, it is recommended that you configure that directory, copy all of your program and data files to a DFS directory using CATCOPY, and then *un-configure* C:\HPW. This will protect against accidental confusion between DFS and HPW directories. Refer to chapter 7 in *Installing and Using HP BASIC/DOS 6.2* for further information.

---

## Running CATCOPY

Once HP BASIC is running you can start the CATCOPY program as follows. Insert the “Manual Examples, LIFINIT, and Selected CSUBs” disk in drive A: and type:

```
LOAD "CATCOPY:DOS,A" 
```

and then press .

The first two screens are informational — they describe what the program does. Read these screens if you wish, and press the  softkey each time to go to the next screen.

The third screen prompts you to identify the *source* disk drive or volume. In the following example, two select code 15 disk drives have been configured:

```
                                CATCOPY DISK UTILITY

Following disks are configured as PC internal mass storage.

                                :,1500,0
                                :,1500,1

Please enter a source drive number from 0 to 1
-
```

Let’s assume that you have configured the PC drives as follows: “:,1500,0” is drive A: and “:,1500,1” is directory C:\HPW on drive C:. If you want to copy files from the latter, type:

```
1 
```

and the following screen will appear:

Do you want to proceed with copy or do you want to do CAT of :,1500,1?

Please press a softkey.

					User 1	Caps	Running
Continue			CAT			Done	

You are now asked whether you want to list a CAT of “:,1500,1”. To do so, press **CAT** (**F5**). The following is an example of a typical listing:

CATCOPY DISK UTILITY

Cat of :,1500,1

:,1500,1

VOLUME LABEL: HPW\_C

FILE NAME	PRO	TYPE	REC/FILE	BYTE/REC	ADDRESS	DATE	TIME
REVID		ASCII	2	256	26	7-Mar-89	16:34
SYSTEM_BA5		SYSTEM	2370	256	28	7-Mar-89	16:38
TEST_PROG		PROG	1	256	2398	16-Mar-89	11:01
DATAFILE01		BDAT	1	256	2399	16-Mar-89	13:34
DATAFILE02		BDAT	1	256	2400	16-Mar-89	13:43
Datafile01		HP-UX	256	1	2401	2-Apr-89	10:11

Do you want to proceed with copy or do you want to do CAT of :,1500,1?

Please press a softkey.

					User 1	Caps	Running
Continue			CAT			Done	



The CAT listing shows all of the files in C:\HPW (“:,1500,1”). The “REVID” file identifies the revision of the measurement coprocessor software in “C:\HPW”. The “SYSTEM\_BA5” file is the HP BASIC 5.0 system file in “C:\HPW”. The remaining files are program and data files — it is these files that you will want to copy to your new system directory. Press **Continue** to proceed.

CATCOPY DISK UTILITY

Please enter the complete destination DOS directory path name,  
for example C:\BLP\PROGS

If the destination DOS directory does not exist, it can be created  
at your option.

Destination DOS directory name?

-

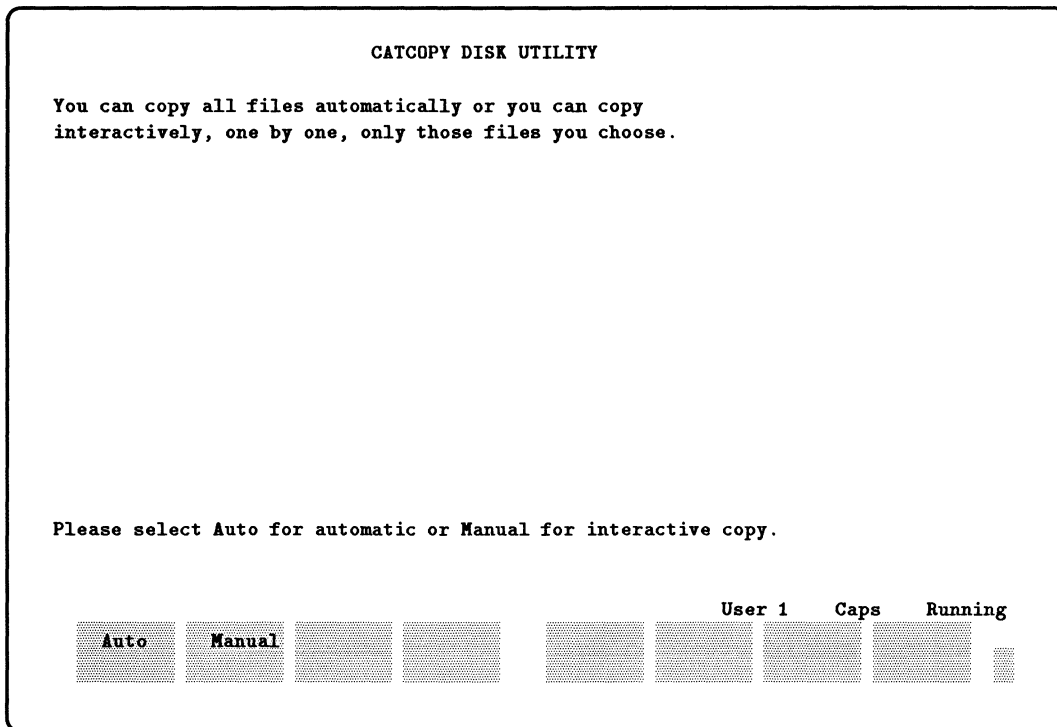
Enter the name of the DFS *destination* directory to which you want to copy your files. (If the directory doesn't already exist, you will be given the option of having CATCOPY create the directory.) For example, type:

C:\BLP\DATA

---

## Choosing the “Auto” or “Manual” Copy Method

Once you have entered a destination directory, the following screen appears:

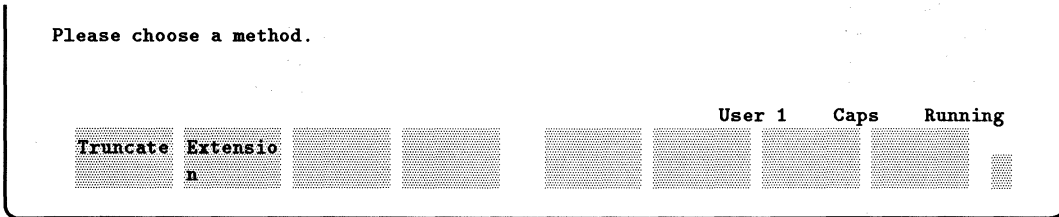


You can choose either **Auto** or **Manual**. The “Auto” method copies all files (except “SYSTEM” files) from the source to the destination directory. Files may include both program and data files. The “Manual” method prompts you to confirm each file before it is copied.

Let’s look at the “Auto” procedure first.

## The “Auto” Copy Method

To select the “Auto” copy method, press **Auto** (F1) in the previous screen. Now you are prompted to select a method for naming the destination file if the source file name is longer than eight characters. (Remember that DFS file names can consist of up to eight characters, plus a “dot extension” of up to three characters.) You can choose either **Truncate** (F1) or **Extension** (F2):



## The “Truncate” Option

If you choose **Truncate**, each file name will simply be truncated to eight characters. (The file name will also be case folded into uppercase characters.) If truncation or case folding results in a file name that already exists, a single-character, alphabetic name extension will be appended. (You won’t get an error message, as you would with the COPY statement.) Press **Truncate** (F1) and the following screen appears:

CATCOPY DISK UTILITY

COPYING files from drive: :,1500,1  
To DOS directory: C:\BLP\DATA

All source file will be copied automatically without your input.  
The old system files will not be copied automatically because they cannot be used with the new measurement coprocessor software.  
If the source LIF file name is longer than 8 characters, the excess characters will be truncated in the destination file name.  
DOS file names are all uppercase. If case folding or truncation of the source file names result in duplicate file names in the destination directory, a letter from A..Z is appended to make the file name unique.

Press **Continue** to start the copying process. The following screen is typical.

CATCOPY DISK UTILITY

COPYING: TEST\_PROG:,1500,1 TO: \BLP\DATA\TEST\_PRO:DOS,C

COPYING: DATAFILE01:,1500,1 TO: \BLP\DATA\DATAFILE:DOS,C

COPYING: DATAFILE02:,1500,1 TO: \BLP\DATA\DATAFILE:DOS,C  
 DOS file \BLP\DATA\DATAFILE:DOS,C exists. Making a unique name.

COPYING: DATAFILE02:,1500,1 TO: \BLP\DATA\DATAFILE.A:DOS,C

COPYING: Datafile01:,1500,1 TO: \BLP\DATA\DATAFILE:DOS,C  
 DOS file \BLP\DATA\DATAFILE:DOS,C exists. Making a unique name.

COPYING: Datafile01:,1500,1 TO: \BLP\DATA\DATAFILE.A:DOS,C  
 DOS file \BLP\DATA\DATAFILE.A:DOS,C exists. Making a unique name.

COPYING: Datafile01:,1500,1 TO: \BLP\DATA\DATAFILE.B:DOS,C  
 Done with copying. Do you want to copy more files?

Please press a softkey.



Note that the “REVID” and “SYSTEM\_BA5” files are *not* copied by the “Auto” procedure. These files are part of the old BASIC system, and are not compatible with Release III.

**The “Extension” Option**

If you choose **Extension** instead of **Truncate**, the ninth and tenth characters of the LIF or HPW file name become an extension of the file name. Again, the file name is case folded. If case folding results in a file name that already exists, a single alphabetic character will be added as the third character in the file name extension. The procedure is essentially the same, but the final screen will appear as follows.

**A-8 Using the CATCOPY Utility to Convert Files**

CATCOPY DISK UTILITY

```
COPYING: TEST_PROG: ,1500,1 TO: \BLP\DATA\TEST_PRO.G:DOS,C
COPYING: DATAFILE01: ,1500,1 TO: \BLP\DATA\DATAFILE.01:DOS,C
COPYING: DATAFILE02: ,1500,1 TO: \BLP\DATA\DATAFILE.02:DOS,C
COPYING: Datafile01: ,1500,1 TO: \BLP\DATA\DATAFILE.01:DOS,C
DOS file \BLP\DATA\DATAFILE.01:DOS,C exists. Making a unique name.
COPYING: Datafile01: ,1500,1 TO: \BLP\DATA\DATAFILE.01A:DOS,C
Done with copying. Do you want to copy more files?
```

As you can see, the “Extension” option fits this particular situation better than the “Truncate” option.

### The “Manual” Copy Method

The “Manual” copy method works essentially like the “Auto” method of the previous section, but you will be prompted whether to copy each file. The following is a typical screen:

Copy REVID: ,1500,1 ?

No skip	Restart			Yes Copy	User 1	Caps	Running
							Done

If you use the “Manual” copy method, you should skip “REVID” and any old “SYSTEM” files such as “SYSTEM\_BA5”.

---

## Exiting CATCOPY

Whatever procedure you follow, when all files are copied you will be prompted to press the **Restart** softkey to begin again (perhaps with a new directory) or **Done** to quit.

## HP BASIC Keyword Differences

---

For the most part, Release III of the measurement coprocessor software (BASIC/DOS 6.2) provides the same keyword functionality as does BASIC/WS 6.2. However, there are a few keywords that function differently for the measurement coprocessor. Also, some keywords have additional functionality.

---

### Keyword Compatibility with BASIC/WS 6.2

The *HP BASIC 6.2 Language Reference* manual gives a detailed description of each keyword for each implementation of BASIC (BASIC/WS, BASIC/DOS, and BASIC/UX). The following table summarizes some of the key differences for porting purposes.

<b>BASIC Keyword</b>	<b>BASIC/WS 6.2 (Series 200/300)</b>	<b>BASIC/DOS 6.2 (Measurement Coprocessor)</b>
ALPHA ON ALPHA OFF	Turn the alpha display on and off. For Series 300, they work only after SEPARATE ALPHA FROM GRAPHICS has been executed.	For VGA or EGA display, same as Series 300. For other displays, SEPARATE ALPHA FROM GRAPHICS and ALPHA ON/OFF are not supported.
AREA	Specifies an area fill color.	Works the same as for BASIC/WS for a VGA color display. For an EGA display, the fill color selection is limited.
BYE	Not applicable.	Exits BASIC to MS-DOS.



BASIC Keyword	BASIC/WS 6.2 (Series 200/300)	BASIC/DOS 6.2 (Measurement Coprocessor)
CAT	Lists the contents of a mass storage directory or provides information on a specified PROG file. (Wildcards may be used.)	Same as BASIC/WS, but adds DFS support. The DFS "header" has four lines:  DIRECTORY: <directory path> LABEL: <drive label or empty> FORMAT: DOS AVAILABLE SPACE: <number of 256-byte blocks>
CONTROL <I/O path>,9	Not applicable.	Used to select buffered or unbuffered mode of I/O path output to a DFS file. For example, the statement "CONTROL @File,9;0" selects unbuffered output for the I/O path "@File". The statement "CONTROL @File,9;1" restores buffered mode, the default.
CONTROL <I/O path>,10	Not applicable.	For buffered output to a DFS file, writing a "1" to this register causes all pending data in the buffer to be written to the disk file. For example, "CONTROL @File,10;1" causes all data to be written to the file assigned to the I/O path "@File".
CONTROL 32	This pseudo select code is used to turn on and off parity checking, system (memory) cache, and processor (instruction) cache.	Not supported for the HP 82300. However, register 2 (FPU enable/disable) and register 3 (instruction cache enable/disable) are supported for the HP 82324.

## B-2 HP BASIC Keyword Differences

<b>BASIC Keyword</b>	<b>BASIC/WS 6.2 (Series 200/300)</b>	<b>BASIC/DOS 6.2 (Measurement Coprocessor)</b>
CONTROL CRT,14	Sets display replacement rule.	Not supported.
CONTROL CRT,18	Sets alpha write-enable mask to a bit pattern.	Not supported. Use SET ALPHA MASK instead.
CONTROL CRT,21	Switches between Series 200 CRT compatibility mode and native bit-mapped mode. Also used to switch between CRT binaries.	Not supported.
COPY	Copies a file from one volume (LIF, HFS, SRM) to another. Also can be used to copy an entire LIF or HFS volume, provided the destination volume is large enough. (Wildcards may be used.)	Works same as for BASIC/WS, but you can also copy files to and from DFS and HPW volumes in the internal PC disk drives. HFS is supported only if the HFS binary is present.
CREATE CREATE ASCII CREATE BDAT CREATE DIR	The first three statements create, respectively, an HP-UX file, an ASCII file, and a BDAT file. In each case, space is pre-allocated on the disk for the file. The files are extensible for SRM or HFS volumes, but not for LIF volumes. The CREATE DIR statement creates a new SRM or HFS directory.	These statements work the same as for BASIC/WS for LIF, SRM, and HFS (if present) volumes. In addition, HP-UX, ASCII, and BDAT files may be created in HPW volumes. For DFS, the CREATE statement creates a DOS file, while the other statements create ASCII files, BDAT files, and new directories. All file types are extensible for DFS.

<b>BASIC Keyword</b>	<b>BASIC/WS 6.2 (Series 200/300)</b>	<b>BASIC/DOS 6.2 (Measurement Coprocessor)</b>
ENTER 19	Not applicable.	Select code 19 is the old measurement coprocessor MS-DOS communication port. ENTER 19 is used by the measurement coprocessor to obtain information from that port (through the "HPW_PIPE" file).
EXECUTE	Not applicable.	Executes an MS-DOS command from BASIC.
GRAPHICS ON GRAPHICS OFF	Turn the graphics display on and off. For Series 300, they work only after SEPARATE ALPHA FROM GRAPHICS has been executed.	For VGA or EGA display, same as Series 300. For other displays, SEPARATE ALPHA FROM GRAPHICS is not supported and, thus, neither are GRAPHICS ON/OFF.
INITIALIZE	Initializes (formats) mass storage media for use as LIF volumes. Also used to initialize RAM volumes.	Works the same as for BASIC/WS for LIF media in external HP-IB disk drives only. Cannot be used to initialize LIF media in internal PC drives (use the LIFINIT utility). In the internal drives, INITIALIZE formats an HPW (virtual-LIF) disk through select code 15. INITIALIZE cannot be used to format a DFS disk (use the MS-DOS "FORMAT" command).
KNOBX KNOBY	Returns the net number of horizontal knob pulses (KNOBX) or vertical knob pulses (KNOBY).	Works the same as for BASIC/WS, except BASIC 2.0/2.1 compatibility (the KNB2_0 binary) is not supported.

#### B-4 HP BASIC Keyword Differences

BASIC Keyword	BASIC/WS 6.2 (Series 200/300)	BASIC/DOS 6.2 (Measurement Coprocessor)
MASS STORAGE IS (MSI)	Specifies the MSVS of the current mass storage volume. For HFS or SRM, can also specify the directory path name. (Wildcards may be used.)	Works the same as for BASIC/WS, but also adds functionality. For HPW volumes, access is through select code 15. For DFS mass storage, the MSI statement specifies the PC drive and directory path name (for example, “\BLP\DATA:DOS,C”) of the current mass storage.
ON/OFF TIMEOUT	Enables/cancels event-initiated branches.	Works the same as for BASIC/WS except for timing differences for select codes 9, 15, 19, and 23 through 26. For these select codes the delay time is device dependent, is usually longer than for BASIC/WS, and may differ between the HP 82300 and HP 82324 coprocessors.
OUTPUT 19	Not applicable.	Select code 19 is the old measurement coprocessor MS-DOS communication port. OUTPUT 19 is used by the measurement coprocessor to send information to that port.
PASS CONTROL	Passes Active Controller capability to a specified HP-IB device.	Works the same as for BASIC/WS <i>for the measurement coprocessor “built-in” HP-IB only.</i> Not supported for PC plug-in HP-IB cards.

<b>BASIC Keyword</b>	<b>BASIC/WS 6.2 (Series 200/300)</b>	<b>BASIC/DOS 6.2 (Measurement Coprocessor)</b>
PERMIT	Modifies the owner, group, or public access permissions of an HFS file or directory. (Wildcards may be used.)	Works the same as for BASIC/WS for HFS (if present). However, PERMIT is implemented slightly differently for DFS. Refer to the <i>HP BASIC 6.2 Language Reference</i> manual for differences.
PRINT LABEL	Gives a name to a mass storage volume.	Not supported on DFS.
PURGE	The PURGE statement deletes files from a volume or directory. For HFS and SRM volumes, PURGE can be used to delete an empty directory from its superior directory. (Wildcards may be used.)	Works the same as for BASIC/WS. In addition, PURGE can be used for DFS mass storage to delete files and empty directories. Note that PURGE cannot delete a DFS file that has its MS-DOS read-only attribute set.
QUIT	Not applicable.	Exits BASIC to MS-DOS.
READIO	This function reads the contents of hardware registers or memory locations directly.	Works the same as for BASIC/WS. However, the use of READIO and WRITEIO requires specific knowledge of the hardware. It is recommended that you use the STATUS and CONTROL statements instead.
READ LABEL	Reads a volume label into a string variable.	Not supported on DFS.
RENAME	Changes a file name or (for HFS or SRM) a directory name. (Wildcards may be used.)	Works the same as for BASIC/WS, but also renames DFS files and directories.

<b>BASIC Keyword</b>	<b>BASIC/WS 6.2 (Series 200/300)</b>	<b>BASIC/DOS 6.2 (Measurement Coprocessor)</b>
SEPARATE ALPHA FROM GRAPHICS	Puts Series 300 computer into Series 200-style separate alpha and graphics display mode (for multiple-plane displays only).	For VGA or EGA display, same as Series 300. Not supported for other displays.
SET CHR	Redefines the bit-pattern used for characters in the current font (on bit-mapped alpha/graphics displays only).	Works the same as Series 300, except that all bits in the bit pattern for a character must be of the same color. Also, SCRATCH ALL does not reset the font as it does on Series 300.
SET PEN	Defines the color for one or more entries in the color map.	With a VGA color display, works same as for Series 300. However, with an EGA display, the color selections are limited.
SET TIME	Resets the time-of-day given by the real-time clock.	With MS-DOS 3.1 and 3.2, SET TIME affects only the "local" BASIC time and MS-DOS time. With MS-DOS 3.3 and above, SET TIME also sets the CMOS, battery-backed clock (the real time clock on the PC).
SET TIMEDATE	Resets the absolute seconds (time and date) given by the real-time clock.	With MS-DOS 3.1 and 3.2, SET TIMEDATE affects only the "local" BASIC time and MS-DOS time. With MS-DOS 3.3 and above, SET TIMEDATE also sets the CMOS, battery-backed clock (the real time clock on the PC).
SOUND	Generates a single tone or multiple tones on the sound generator of an HP-HIL interface.	Not supported for the measurement coprocessor. (Use BEEP instead.)

<b>BASIC Keyword</b>	<b>BASIC/WS 6.2 (Series 200/300)</b>	<b>BASIC/DOS 6.2 (Measurement Coprocessor)</b>
STATUS <I/O path>,9	Not applicable	Used to determine the mode of output to a DFS file through an I/O path. A returned value of "1" indicates buffered mode and a value of "0" indicates unbuffered mode.
SYSTEM\$ ("MASS MEMORY")	Returns X000YZ0000000000 where:  X = Number of internal disk drives.  Y = Number of initialized EPROM cards.  Z = Number of bubble memory cards.  (If Y or Z exceeds 9, an asterisk appears.)	Always returns 0000000000000000
SYSTEM\$ ("MASS STORAGE IS")	Returns the mass storage volume specifier (MSVS) of the current MASS STORAGE IS device as it appears in the CAT heading.	For DFS, returns "<dir_path>:DOS,<dr_letter>". Otherwise, same as BASIC/WS.
SYSTEM\$ ("SERIAL NUMBER")	Returns one of the following:  Bytes 4-14 of an ID PROM (if present).  An encoded serial number from an HP-HIL ID module (if present).  Otherwise, returns null string.	Returns:  An encoded serial number from an HP-HIL ID module (if present).  Otherwise, returns the string "11111111".

<b>BASIC Keyword</b>	<b>BASIC/WS 6.2 (Series 200/300)</b>	<b>BASIC/DOS 6.2 (Measurement Coprocessor)</b>
TRANSFER	Initiates unformatted I/O transfers, which can take place concurrently with program execution.	Generally works the same as for BASIC/WS. However, overlapped transfers on DFS, HPW, or emulated LIF disks are not supported. Also, serial and parallel transfers are not supported.
WAIT	Causes the computer to wait approximately the specified number of seconds before executing the next statement.	Works the same as for BASIC/WS. However, you can change the mode of WAIT for compatibility with Releases I and II as described in the next section.
WRITEIO	This function writes directly to hardware registers or memory locations.	Works the same as for BASIC/WS. However, the use of READIO and WRITEIO requires specific knowledge of the hardware. It is recommended that you use the STATUS and CONTROL statements instead.

---

**Caution**



It is possible to damage your measurement coprocessor by writing to certain memory locations with WRITEIO. Hewlett-Packard cannot be held liable for any damages caused by improper use of this feature.

---



---

## Changing the Mode of the WAIT Statement

The WAIT statement has been changed in Releases II-A and III for compatibility with BASIC/WS — “WAIT 1” now gives approximately a 1 second delay. In Releases I and II the WAIT statement provided a delay only 80 percent of the specified time.

For example, the statement:

```
WAIT 10
```

causes a delay of approximately 10 seconds for Release II-A, Release III, and for BASIC/WS, but a delay of only 8 seconds for Releases I and II.

If you have written programs for Release I or II that contain WAIT statements, you may want to change these statements to give the desired delay period. However, if you have several programs with lots of WAIT statements, you may find it easier to change the mode of the WAIT statement to make it compatible with Releases I and II. Let's look at how this is done.

By default, the Release III WAIT statement gives an “accurate” delay (“WAIT 1” = 1 second). To change this to an “80 percent” delay (“WAIT 1” = 0.8 second), execute the following BASIC statement:

```
WRITEIO 9826,-295;1
```

To return to the default mode (“WAIT 1” = 1 second), execute:

```
WRITEIO 9826,-295;0
```

or reboot the measurement coprocessor.

To check the current mode of WAIT, execute:

```
READIO (9826,-295)
```

The value returned indicates the mode (“0” = “accurate” and “1” = “80 percent”).

## Porting Programs That Use CSUBs

---

The measurement coprocessor Release III software (HP BASIC/DOS 6.2) is compatible with most CSUBs (compiled subprograms) created with the HP BASIC 6.0 CSUB Utility for BASIC 6.x. However, there are three porting issues that you must consider if you are porting programs that use CSUBs:

- Releases I, II, and II-A of the measurement coprocessor software all provided BASIC 5.x. *Thus, if you are porting from one of these versions, any CSUBs that you created for BASIC 5.x must be relinked using the HP BASIC 6.0 CSUB Utility.* (This is also true of programs being ported from workstation BASIC 5.x, or any earlier version.)
- If you are porting a program from HP BASIC/WS 6.x, most CSUBs will be compatible with HP BASIC/DOS 6.2. However, there are exceptions to this general rule. *If a CSUB attempts to access the hardware directly (for example, graphics memory), it will have to be modified for the measurement coprocessor.* For this reason, selected graphics CSUBs are supplied with HP BASIC/DOS 6.2.
- If you are using any of the graphics CSUBs (BPLOT, GDUMP\_C, GDUMP\_R, and PCBLOCK) or any of the Multicom CSUBs (BLPLIB, POPLIB, ADVLIB, and 123LIB) provided with BASIC/DOS, make sure that you use the versions of these CSUBs shipped with BASIC/DOS 6.2, not an earlier version.

---

## Relinking CSUBs

A CSUB is a routine that is written in either Pascal or assembly language on a Pascal workstation and transformed into a routine that is callable as a subprogram from BASIC. If a CSUB has been created for HP BASIC 5.x or an earlier version of BASIC, it must be **relinked** using the HP BASIC 6.0 CSUB Utility before you can use it with HP BASIC/DOS 6.2. For information about creating and relinking CSUBs, refer to the *HP BASIC 6.0 CSUB Utility* manual.

---

### Note



BASIC/DOS 6.2 and above (and BASIC/WS 6.0 and above) have been changed internally so that you will no longer have to relink your CSUBs for subsequent versions of BASIC. For example, you will have to relink your BASIC 5.x CSUBs to work with BASIC 6.x, but you won't have to relink them again for the next release of BASIC.

---

---

## CSUB Errors

If your program uses a CSUB that is not compatible with the measurement coprocessor, you will receive an HP BASIC error message in the range 369 through 398. Errors in this range indicate a CSUB Pascal error (probably because of a hardware dependency), for example:

```
ERROR 389 IN 4 CSUB Pascal error -11
```

You can obtain information about Pascal errors in the *Pascal Workstation System* manual.

---

## CSUBs Provided with Release III

The graphics CSUBs “BPLOT”, “GDUMP\_C”, “GDUMP\_R”, and “PCBLOCK” access the hardware directly. Thus, special versions of these CSUBs are provided with Release III (BASIC/DOS 6.2). They are found on the “Manual Examples, LIFINIT, and Selected CSUBs” disk. You must use the correct versions of these CSUBs in your programs. The BASIC/WS versions and earlier BASIC/DOS versions won’t work with BASIC/DOS 6.2.

If you are porting a program that uses one of the graphics CSUBs from Series 200/300, or from an earlier version of the measurement coprocessor, you will have to replace the CSUBs with the Release III versions. To see how to do this, let’s look at an example. Suppose the following program was created, and the “BPLOT” CSUBs “Bload” and “Bstore” were loaded, on an HP 9000 Series 300 BASIC Workstation.

```
10 ! Bstore with VGA (Pixel_ratio = 1.0)
20
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
140 CSUB Bload(INTEGER A(*),Xpixels,Ypixels,OPTIONAL INTEGER Rule,REAL Xstart,Ystart)
150 CSUB Bstore(INTEGER A(*),Xpixels,Ypixels,OPTIONAL INTEGER Rule,REAL Xstart,Ystart)
```

The same program will run on the measurement coprocessor once you have done the following:

1. Boot HP BASIC/DOS 6.2 on the measurement coprocessor.
2. LOAD the program (from a LIF flexible disk, for example).
3. Remove the Series 300 CSUBs by executing the following statement from the HP BASIC command line:

```
DELSUB Bload,Bstore 
```

4. Insert the "Manual Examples, LIFINIT, and Selected CSUBs" disk in drive A:.

5. Load the Release III versions of the same CSUBs by executing:

```
LOADSUB ALL FROM "BPLOTT:DOS,A" 
```

The procedure is the same for programs using "GDUMP\_C" or "GDUMP\_R", except the HP BASIC statements are:

```
DELSUB Gdump_colored 
```

and then:

```
LOADSUB ALL FROM "GDUMP_C:DOS,A" 
```

or

```
DELSUB Gdump_rotated 
```

and then:

```
LOADSUB ALL FROM "GDUMP_R:DOS,A" 
```

In addition to the graphics CSUBs, four Multicom CSUBs (BLPLIB, POPLIB, ADVLIB, and 123LIB) are provided with the measurement coprocessor. If you are using Multicom, make sure that you use the versions of these CSUBs shipped with BASIC/DOS 6.2, not an earlier version. (For further information about Multicom, refer to *Integrating HP BASIC With MS-DOS Applications*.)



**HP Part Number**  
**82324-90005**

**Printed in U.S.A. E0791**



**82324-90607 Manufacturing Number**