BASIC 3.0
User's Guide
for the HP 9000 Series 200 Computers

Manual Part No. 98613-90040

Hewlett-Packard Company
3404 East Harmony Road, Fort Collins, Colorado 80525

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

May 1984...First Edition

November 1984 Update

December 1984...First Edition with update merged

March 1985...Update

Warranty Statement

Hewlett-Packard products are warranted against defects in materials and workmanship. For Hewlett-Packard Fort Collins Systems Division products sold in the U.S.A. and Canada, this warranty applies for ninety (90) days from the date of delivery.* Hewlett-Packard will, at its option, repair or replace equipment which proves to be defective during the warranty period. This warranty includes labor, parts, and surface travel costs, if any. Equipment returned to Hewlett-Packard for repair must be shipped freight prepaid. Repairs necessitated by misuse of the equipment, or by hardware, software, or interfacing not provided by Hewlett-Packard are not covered by this warranty.

HP warrants that its software and firmware designated by HP for use with a CPU will execute its programming instructions when properly installed on that CPU. HP does not warrant that the operation of the CPU, software, or firmware will be uninterrupted or error free.

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.

* For other countries, contact your local Sales and Support Office to determine warranty terms.

Table of Contents

Chapter 1: Introduction to BASIC

Chapter 2: Loading BASIC

Chapter 3: Backing Up BASIC

Chapter 4: Using Your Keyboard With BASIC

Chapter 5: Working With BASIC

Chapter 6: Talking To Peripherals With BASIC

Chapter 7: Special BASIC Features

Chapter 8: Configuring Your BASIC System

Appendix A: Task Reference

Introduction to BASIC

Chapter 1

Welcome to your *BASIC 3.0 User's Guide*! This guide introduces you to the BASIC 3.0 language system as it runs on your HP 9000 Series 200 computer. You should read this guide and perform the prescribed tasks **after** your computer system is installed and **before** you dig into the BASIC 3.0 manuals for detailed information.

Installing Your Computer

If you have not yet installed your computer and configured your system, do not continue with this book now. Refer to the *Installation Guide* which came with your computer and perform the procedures required to install your system. After your system is installed, return to this guide.

What Is The BASIC Language System?

Just as we have a common set of words, or vocabulary, for communicating with one another, the computer has a set of words it recognizes and acts on to perform your operating and programmed tasks. Operating tasks include keyboard, display, and disc drive controls. Programmed tasks include input/output operations, decision making, and program debugging. The computer's set of words is called its language system. This guide describes operating your computer with the BASIC language system. BASIC is just that: an easy to learn, yet powerful language for both the keyboard and programming operations.

As with other programming languages, the BASIC language system assigns a unique "keyword" from its vocabulary to each computer instruction or task. For example, to start running a program in BASIC, you can either press the "run" key on the keyboard or type in the keyword `RUN` and press the "execute" key on the keyboard. To list the files on a disc, you type in the keyword `CAT` and press the "execute" key. (Don't worry about your keyboard labels now; "run" and "execute" are key functions, not labels, in this example.) The language system interprets each key pressed or keyword executed and performs a complex set of internal instructions to accomplish the task.

How This Guide Is Organized

This guide has seven chapters besides the one you're now reading. It also has two appendices, a glossary of terms, and a subject index. This guide is **task-oriented**; you should read the chapters in order and perform the procedures and examples as you go. The following paragraphs describe the organization and contents of the rest of this guide.

Note

If you are an experienced BASIC user, you may want to refer to Appendix A, Task Reference, for condensed versions of procedures that are described in detail in the body of this guide.

Chapter 2 - Loading BASIC. This chapter explains how to load the BASIC language system into your computer. This procedure is also called "booting". Proper care and handling of flexible discs is also described here.

Chapter 3 - Backing Up BASIC. This chapter tells you how to create copies of your BASIC system discs. This ensures that you can re-boot the BASIC system if your working discs are lost, damaged, or erased.

Chapter 4 - Using Your Keyboard With BASIC. This chapter describes the three Series 200 keyboards and their operation with BASIC.

Chapter 5 - Working With BASIC. This chapter will get you started programming in BASIC. You will write, run, edit, store, and load a simple program. You will also learn and practice arithmetic operations, as well as learn how to handle errors.

Chapter 6 - Talking To Peripherals With BASIC. This chapter describes in detail how your computer communicates with disc drives and printers.

Chapter 7 - Special BASIC Features. This chapter describes display enhancements, graphics, and calculating features of your BASIC computer.

Chapter 8 - Configuring BASIC. This chapter contains techniques to help you configure and store your BASIC system. You will learn about binary (BIN) files and commands used to configure BASIC. You will also learn how to set up an autostart file.

Appendix A - Task Reference. This appendix provides condensed procedures of BASIC operations that are described in detail in the body of the guide.

Appendix B - BASIC Documentation. This appendix lists all the BASIC documentation and part numbers.

Glossary of Terms. This is an alphabetical list of commonly used terms contained in this guide along with their definitions.

Subject Index. This is an alphabetical list of subjects discussed in this guide and the page numbers where they're found.

Loading BASIC

Chapter 2

The first step toward making your computer system operational is to add some software to your hardware. This chapter describes how to load the BASIC language system into memory. The BASIC language system, other language and operating systems, and stand-alone application programs are commonly called **system programs**.

There are two ways to load your BASIC system. The easiest and most common method is used whenever the BASIC system is the only system that your computer can access. Usually this means that your computer accesses one flexible disc drive, and the BASIC system disc is in that disc drive.

The other method for loading your BASIC system should be used whenever you have multiple systems that your computer can access. This technique allows you to select which system you want to load. Both ways are described in this chapter.

What Does BASIC Do?

The best way to demonstrate what the BASIC system does is to try using your computer without it.

Try It Yourself

Turn your computer on. Try typing some keys. Rotate the cursor control wheel if you have one. If you're lucky, you might stumble onto a few special keys that are operational, but for the most part, there is very little your computer can do. At the bottom of the screen you'll see the message:

```
SEARCHING FOR A SYSTEM
```

or

```
UNABLE TO FIND SYSTEM
RESET TO RETRY
```

The computer is telling you that it desperately needs a system program; without one, it is helpless.

The BASIC system is a master set of instructions that takes care of all the minute computing details we human beings would rather not worry about. As you saw in the demonstration above, one of the functions of the BASIC system is to define the operation of each key on the keyboard. The BASIC system also manages the exchange of information between the computer and its peripherals, refreshes the display on the CRT, reserves space for programs in random access memory (RAM), and performs several other functions too numerous to mention.

Besides these resource management duties, the BASIC system also supports a **human interface** that makes it easy for people to communicate with the computer and to design and run programs. The BASIC language system supports the BASIC programming language. By ''supports'' we mean that the BASIC language system interprets the BASIC language for the processor.

People and computer processors don't speak the same language. BASIC, although not exactly a ''natural'' language, is fairly easy for people to learn. It uses simple mathematical expressions and common English words in its statements. Processors, on the other hand, find BASIC far too complex for their weak intellects. Processors prefer **machine language**, with instructions expressed in terms of binary digits: ones and zeroes. But asking people to learn a hundred or so instructions that look like 0001110110011110 is asking a bit much.

So the BASIC language system was invented to interpret between people and processors. The BASIC language system intercepts BASIC language statements, translates them into their equivalent machine language instructions, and hands these to the processor. This way, processors can speak one language, and people can speak another.

The Boot ROM

The process of loading the BASIC language system into memory is called "booting" BASIC. As you might guess, the instructions for booting BASIC are contained in the boot ROM (Read-Only Memory).

There are four boot ROMs supported by the BASIC language system: 1.0, 2.0, 3.0, and 4.0. You have **one** of these boot ROMs in your computer. Boot ROMs can be divided into the following two groups:

1.0 and 2.0 - "earlier" boot ROMs
3.0 and 4.0 - "later" boot ROMs

The visible difference between the two groups of boot ROMs is in the power-up display. The later boot ROMs (3.0 and 4.0) provide more information than the earlier boot ROMs (1.0 and 2.0). The following paragraphs describe the earlier boot ROMs. The rest of this guide, beginning with "Special Keys", assumes you have the later boot ROMs except as noted.

Boot ROMs 1.0 and 2.0 (Earlier Boot ROMs)

Earlier boot ROMs do not display their revision number at powerup. If your computer displays `BOOTROM 3.0` or a later revision at powerup, you can skip this section; continue with "Special Keys". If your computer does not display a boot ROM revision at powerup, your computer's boot ROM operation is explained here.

Earlier boot ROMs first perform a memory test and keyboard test. The initial display is:

```
MEMORY TEST IN PROGRESS
```

Then the amount of memory available is displayed:

`851536 AVAILABLE BYTES` (number of bytes may vary)

Any ROM cards and the internal right-hand disc drive are finally scanned for an operating system.

If a system is found, either on a ROM card or on disc, the computer loads it and displays a "READY" message. If more than one system is found, the computer allows you to select one. For example:

```
WHICH SYSTEM?
B H
```

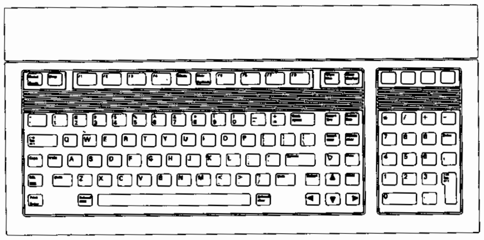
In this case the computer found two systems, BASIC and HPL. The computer will wait about 10 seconds for you to indicate your choice by pressing the appropriate key. If you press another key, the computer will just beep and continue waiting. If an appropriate key isn't pressed in time, the first system listed (BASIC in this case) will be automatically loaded.

The boot ROM may encounter problems when testing and trying to load a system. The error messages for earlier boot ROMs are listed in your computer's *Installation Guide*.

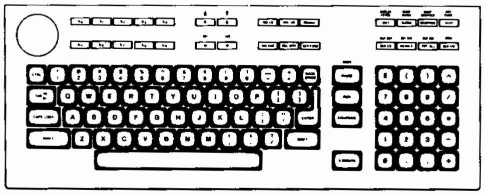
Special Keys

There are a few keys that have special functions during the self-test and booting process. The functions of these keys change as soon as a system is loaded. The special keys are defined in the following paragraphs.

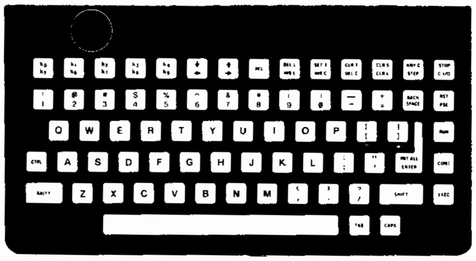
You have one of the following keyboards: HP 46020A, HP 98203B, or HP 98203A. Your keyboard and its operation are described in Chapter 4. For now, identify your keyboard from the following illustration and continue below.



HP 46020A



HP 98203B



HP 98203A

The Reset Key

HP 46020A - Reset (Shift-Reset)
HP 98203B - RESET (SHIFT-PAUSE)
HP 98203A - RST (SHIFT-RST)

Pressing the reset key has the same effect as turning the machine off and then on again. It aborts the booting or self-test process and starts it all over again. You can press the reset key anytime before a system is booted. After the system program takes control, the reset key and all other keys are re-defined.

Try It Yourself

Turn your computer off and then on again. Quickly press the reset key. Notice that the self-test begins listing components on the display again, indicating that the test has restarted. Following that, the message:

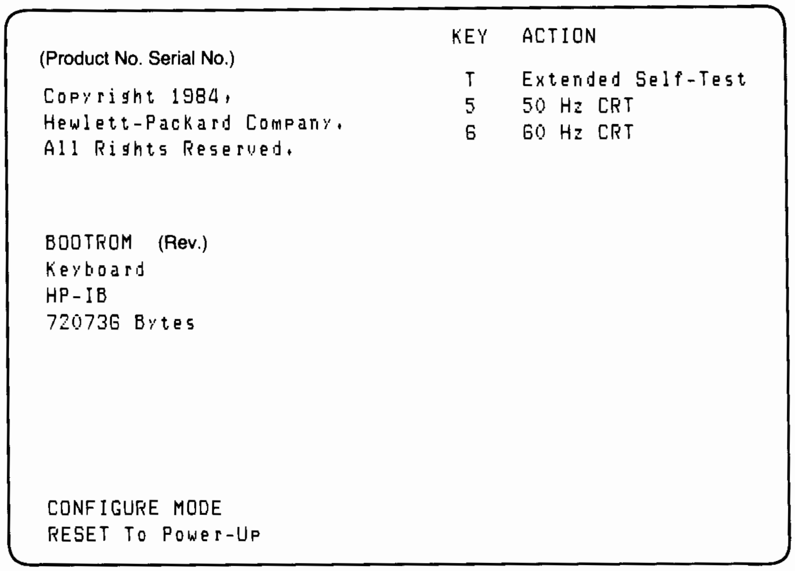
```
   SEARCHING FOR A SYSTEM
```

reappears at the bottom of the screen, signaling that the booting process is underway.

The Control-C Key

All keyboards have the CTRL and C keys.

When you hold down CTRL and press C during either the self-test or booting process, the computer stops to allow you to change the CRT Hz setting. The display you see after pressing CTRL-C looks like this:



```
                                        KEY   ACTION
(Product No. Serial No.)
                                         T    Extended Self-Test
Copyright 1984,                          5    50 Hz CRT
Hewlett-Packard Company,                 6    60 Hz CRT
All Rights Reserved.


BOOTROM  (Rev.)
Keyboard
HP-IB
720736 Bytes





CONFIGURE MODE
RESET To Power-Up
```

The system waits five minutes for you to press one of the keys listed in the upper-right portion of the screen; after five minutes, the computer automatically RESETs. Type 5 to set the CRT to 50 Hz or 6 to set it to 60 Hz. Ignore the T (Extended Self-Test) key–the computer automatically performs an extended self-test after you type your selection.

Try It Yourself

Press the reset key to restart the self-test, then press CTRL-C. Now type 5 to configure the CRT to 50 Hz and see how the display looks. Then press the reset key and CTRL-C again. This time type 6 to select 60 Hz. Use the setting that provides the better display. At power-up the setting defaults to the frequency selected by a jumper or switch in your computer. Call your local HP representative if you want the jumper changed.

The Enter Key

HP 46020A - Return
HP 98203B - ENTER
HP 98203A - ENTER

The enter key performs a number of functions. As the computer proceeds from its self-test to the booting process, the function of the enter key changes.

During the self-test, the enter key is only used when an error is detected in one of the components. For details on this function of the enter key, refer to the chapter entitled "Reading the Self-Test" in your computer's *Installation Guide.*

During the booting process, the enter key is used to suspend (pause) the search for system programs. Pressing the enter key a second time resumes the search. This is particularly useful when you have several systems that the computer can find. There is a way (discussed later in this chapter) to list all accessible systems on the screen. When there are more systems to list than can fit on the screen, you can use the enter key to freeze the display before the systems at the top are overwritten.

Try It Yourself

Turn your computer on, or press the reset key. Wait for the message:

```
SEARCHING FOR A SYSTEM (ENTER To Pause)
```

to appear at the bottom of the screen, signaling that the booting process has begun. Immediately press the enter key, and this message will change to:

```
SEARCHING PAUSED (ENTER To Continue)
```

The computer has now stopped searching for a system to boot. Press the enter key again, and the original message will return.

Your BASIC Discs

Your BASIC language system comes on 3½-inch flexible discs (microdiscs) or 5¼-inch flexible discs (mini-discs). The next several pages tell you how to use and care for your flexible discs. The first section is applicable to 3½-inch discs; the second section tells you about your 5¼-inch discs.

CAUTION

IF YOU ARE UNFAMILIAR WITH DISCS AND DISC HANDLING PROCEDURES, READ THE FOLLOWING INFORMATION CAREFULLY. OTHERWISE, YOU MAY DAMAGE YOUR DISCS.

The flexible disc is a thin piece of plastic enclosed in a special plastic jacket. The disc is covered with a thin oxide coating on which your program and data information are electronically stored.

When you insert the disc in the drive and close the door, the drive is ready to read information from or write information onto the disc. When the computer requests a read or write, the disc spins at a constant rate, like a phonograph record. The light on the disc drive indicates when reading or writing is taking place. Do not attempt to remove the disc when the light is on. Be sure to use only discs supplied or approved by HP. Other discs may not be of adequate quality or may damage the drive.

Unless you have a hard disc connected to your computer, you'll find that handling, inserting and removing flexible discs become an integral part of your daily routine. Even if you have a hard disc, you must deal with flexible discs initially (BASIC comes on flexible disc), and from time to time after that.

Since flexible discs represent a great deal of your money (the BASIC language system and other software you buy) and your time (the programs that you write), and contain your valuable records (data that you maintain), it is a very wise idea to treat them with considerable respect. This section tells you how to (and how not to) handle flexible discs. It also shows you how to insert and remove discs from your disc drive, which is no obvious matter if you have never done it before.

The following sections describe the handling and use of 3½-inch microdiscs and 5¼-inch mini-discs. Read only the section that pertains to your disc size.

3½-inch Micro-disc Handling and Use

This section describes the handling and use of 3½-inch micro-discs. If you're using 5¼-inch mini-discs instead, skip to the section titled "5¼-inch Mini-disc Handling and Use".

Handling 3½-inch Micro-discs

The flexible micro-disc is basically maintenance free, but is delicate and **must be handled carefully**. A good rule of thumb is to treat your disc as you would a valuable record album.

Here are some specific Do's and Don'ts to avoid loss of data or damage to your discs:

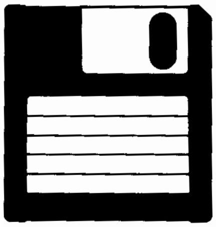
CAUTION

EVEN A LITTLE CARELESSNESS IN DISC HANDLING CAN DRAMATICALLY REDUCE THE LIFE OF A DISC.

Back Up Discs

There is always a chance of losing data anytime you are using your computer. There are many possible causes for this: a programming bug, operator error, power failure, hardware failure, or disc media failure from wearout, damage or contamination. **The only sure protection against data and program loss is to back up your discs**. Backing up discs is covered in Chapter 3.

Slide Disc Guard Over Media Window When Not in Use

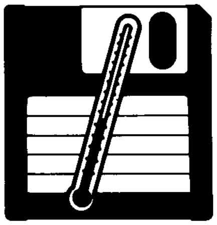


This is the single most important thing to remember about handling your disc. The disc guard protects the media against dust, fingerprints and scratches. **Never attempt to blow dust from the disc**. The disc's self-cleaning mechanism will remove most small particles from the media. When not in use the disc should be stored upright in a dust-free container, such as the box your discs were shipped in.

Operate Your Computer in a Clean Environment

Airborne contaminants and particles accidentally dropped onto the disc will cause your disc to wear out prematurely. Unreliable data storage and retrieval operations may also occur. Some of the most common contaminants are **dust, smoke, ashes, eraser crumbs and food particles. Never attempt to blow small particles from the disc.** Chemical vapors may also cause premature wearout.

Maintain Proper Temperature and Humidity



The proper operating range is 10°C (45°F) to 45°C (115°F) and 20% to 80% relative humidity. While temperature is usually easy to control, you may need to make special provisions to keep the humidity in the proper range. Although the disc will operate outside the normal humidity range, it will wear out more quickly and have a higher error rate.

Avoid Magnetic Fields

Data is stored on the disc magnetically, and can be erased by an external magnetic field. Avoid placing the disc near power transformers, magnets, large disc memories or motors.

Remove the Disc from the Drive When Not in Use

Remove the disc completely from the drive when you are through using it.

Use a Felt Tip Pen to Label Your Disc

Use a soft felt tip pen to label your disc, and be careful to write only in the label area. Using a hard tip pen, such as a ball point, can damage the media.

Don't Touch the Surface of the Disc

Be careful not to touch the media surface that is exposed through the media window. The thickness of a fingerprint is enough to lift the head off the disc and cause errors. The oil in a fingerprint will also collect dust and cause a disc to wear out prematurely.

Don't Bend or Fold the Disc

The disc is flexible but will not operate if creased. Using ball point pens, rubber bands, paper clips, etc., can crease the disc.

Don't Clean Your Disc

The inside surface of the disc jacket is covered with a special material that cleans the disc as it rotates in the drive. Any other method of cleaning may damage the media and cause data loss. **If a disc becomes dirty or scratched, immediately transfer the data to a new disc and dispose of the old one.**

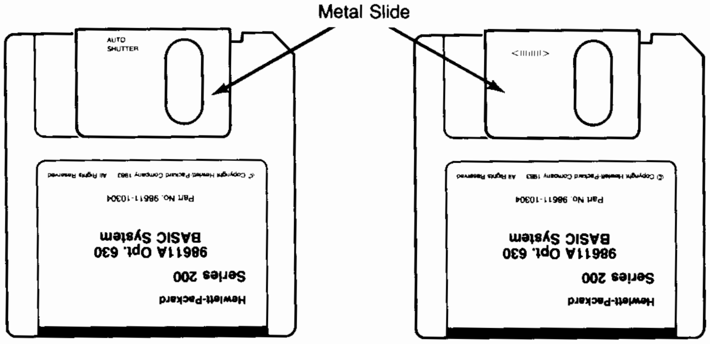
CAUTION

IF YOUR DISC DRIVE EVER DESTROYS THE MEDIA ON A DISC, STOP USING THE DRIVE UNTIL IT CAN BE SERVICED. THIS IS ESPECIALLY IMPORTANT, AS CONTINUED USE OF THE DRIVE WILL DESTROY MORE MEDIA. IMMEDIATELY CALL YOUR NEAREST HP SALES AND SERVICE OFFICE

Inserting and Removing 3½-inch Micro-discs

In this section, you will learn how to insert and remove 3½-inch micro-discs to and from your disc drive. If you're using 5¼-inch mini-discs, skip to the section titled "5¼-inch Mini-disc Handling and Use".

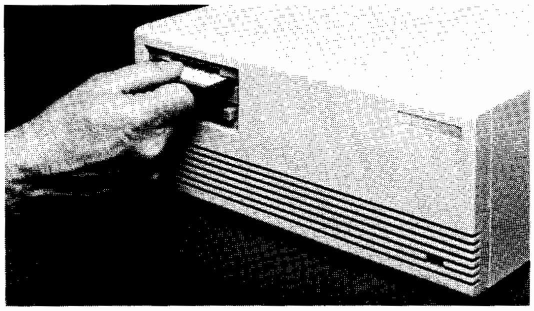
Micro-discs come in two varieties: "auto shutter" and "manual shutter." To find out which type you have, look at the metal slide. If the words "AUTO SHUTTER" appear just above the HP logo, you have an auto shutter disc. If a two-directional arrow appears, you have a manual shutter disc. See the following illustrations.



Auto Shutter Disc **Manual Shutter Disc**

Inserting Auto Shutter Micro-discs into Your Disc Drive

To insert an auto shutter disc, make sure the disc drive is turned on and slide the disc into the disc drive, label side up and metal slide facing the disc drive (try it!).



Inserting 3½-inch Micro-disc

The drive should completely "swallow" the disc and flash its light to signal acceptance. If the drive rejects the disc and pushes it back out at you, follow these steps:

1. Remove the disc and make sure you inserted it label side up. Also check that the disc drive is turned on. If you inserted the disc incorrectly, try it again.
2. If you inserted the disc correctly but the drive still refuses it, move the metal slide to the left, exposing the media. Reinsert the disc.

Note

When possible, turn the disc drive on before inserting a disc.

Inserting Manual Shutter Micro-discs into Your Disc Drive

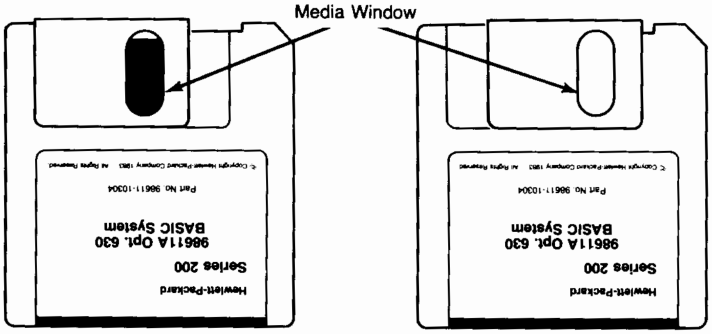
To insert manual shutter micro-discs into your disc drive, follow this procedure (try it!):

1. Make sure your disc drive is turned on.
2. Move the metal slide all the way to the left, exposing the media.
3. Insert the disc into the drive, label-side up and metal end facing the disc drive.

The drive should completely "swallow" the disc and flash its light to signal acceptance. If it pushes the disc back out at you, check that the label is facing up and the drive is turned on. Re-insert the disc.

Removing 3½-inch Micro-discs

To remove a 3½-inch disc from the disc drive, make sure the disc drive is turned on and press the disc eject button in the lower-right corner of the drive (try it!) The drive will immediately eject the disc. Close the metal slide over the media window if it is not already covered.



Media Exposed **Media Covered**

5¼-inch Mini-disc Handling and Use

This section describes the handling and use of 5¼-inch mini-discs. If you're using 3½-inch micro-discs instead, go back to the section titled "3½-inch Micro-disc Handling and Use".

Handling 5¼-inch Mini-discs

The flexible mini-disc is basically maintenance free, but is delicate and **must be handled carefully**. A good rule of thumb is to treat your disc as you would a valuable record album. Here are some specific Do's and Don'ts to avoid loss of data or damage to your discs:

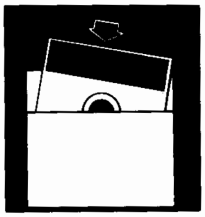
CAUTION

EVEN A LITTLE CARELESSNESS IN DISC HANDLING CAN DRAMATICALLY REDUCE THE LIFE OF A DISC.

Back Up Discs

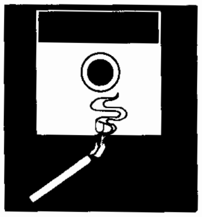
There is always a chance of losing data anytime you are using your computer. There are many possible causes for this: a programming bug, operator error, power failure, hardware failure, or disc media failure from wearout, damage or contamination. **The only sure protection against data and program loss is to back up your discs**. Backing up discs is covered in Chapter 3.

Return Disc to Storage Envelope When Not in Use



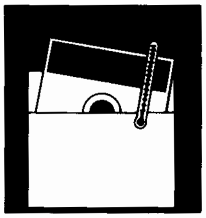
This is the single most important thing to remember about handling your disc. The storage envelope protects the media against dust, fingerprints and scratches. **Never attempt to blow dust from the disc**. The disc's self-cleaning mechanism will remove most small particles from the media. When not in use the disc should be stored upright in a dust-free container, such as the box your discs were shipped in.

Operate Your Computer in a Clean Environment



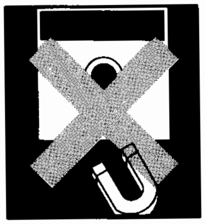
Airborne contaminants and particles accidentally dropped onto the disc will cause your disc to wear out prematurely. Unreliable data storage and retrieval operations may also occur. Some of the most common contaminants are **dust, smoke, ashes, eraser crumbs and food particles. Never attempt to blow small particles from the disc.** Chemical vapors may also cause premature wearout.

Maintain Proper Temperature and Humidity



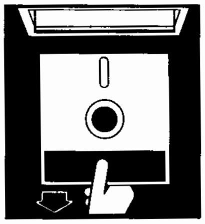
The proper operating range is 10°C (45°F) to 45°C (115°F) and 20% to 80% relative humidity. While temperature is usually easy to control, you may need to make special provisions to keep the humidity in the proper range. Although the disc will operate outside the normal humidity range, it will wear out more quickly and have a higher error rate.

Avoid Magnetic Fields



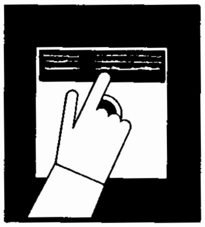
Data is stored on the disc magnetically, and can be erased by an external magnetic field. Avoid placing the disc near power transformers, magnets, large disc memories, or motors.

Remove the Disc from the Drive When Not in Use



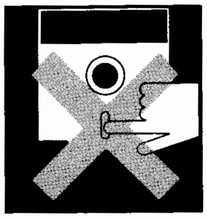
Remove the disc completely from the drive when you are through using it.

Use a Felt Tip Pen to Label Your Disc



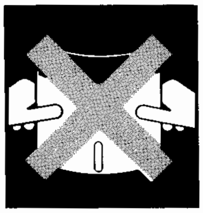
Use a soft felt tip pen to label your disc, and be careful to write only in the label area. Using a hard tip pen, such as a ball point, can damage the media. If possible, write on the large labels provided with your discs BEFORE applying them to the disc.

Don't Touch the Surface of the Disc



Be careful not to touch the media surface that is exposed through the media window. The thickness of a fingerprint is enough to lift the head off the disc and cause errors. The oil in a fingerprint will also collect dust and cause a disc to wear out prematurely.

Don't Bend or Fold the Disc



The disc is flexible but will not operate if creased. Using ball point pens, rubber bands, paper clips, etc., can crease the disc.

Don't Clean Your Disc

The inside surface of the disc jacket is covered with a special material that cleans the disc as it rotates in the drive. Any other method of cleaning may damage the media and cause data loss. **If a disc becomes dirty or scratched, immediately transfer the data to a new disc and dispose of the old one.**

CAUTION

IF YOUR DISC DRIVE EVER DESTROYS THE MEDIA ON A DISC, STOP USING THE DRIVE UNTIL IT CAN BE SERVICED. THIS IS ESPECIALLY IMPORTANT, AS CONTINUED USE OF THE DRIVE WILL DESTROY MORE MEDIA. IMMEDIATELY CALL YOUR NEAREST HP SALES AND SERVICE OFFICE.

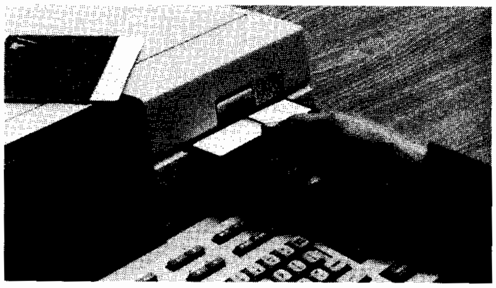
Inserting and Removing 5¼-inch Mini-discs

This section describes how to insert and remove 5¼-inch mini-discs to and from your disc drive. If you're using 3½-inch micro-discs instead, go back to the section titled "3½-inch Micro-disc Handling and Use".

Inserting 5¼-inch Mini-discs into Your Disc Drive

To insert a mini-disc into your disc drive, follow this procedure (try it!):

1. Make sure your disc drive is turned on.
2. Open the drive door by lifting the door handle. Check that there is not already a disc in the drive.
3. Insert the disc into the drive, label-side up with the media window facing the drive.



Inserting a 5¼-inch Mini-disc

4. Close the drive door. If the door will not close, push the disc farther into the drive and try again.

CAUTION

IF YOU ACCIDENTALLY INSERT ANOTHER DISC WHEN ONE IS ALREADY IN THE DRIVE, REMOVE THE BOTTOM DISC FIRST. OTHERWISE, THE READ/WRITE HEADS COULD BE DAMAGED.

Note

When possible, turn the disc drive on before inserting a disc.

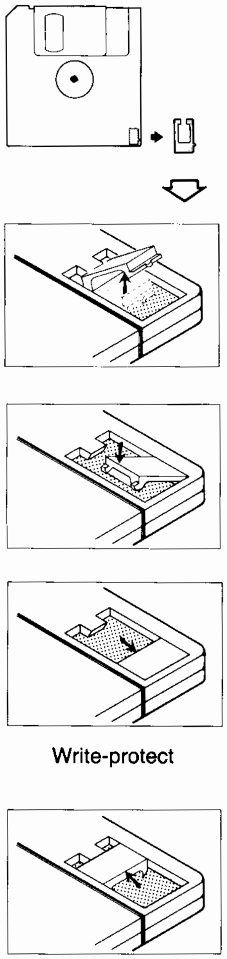
Removing a 5¼-inch Mini-disc from Your Disc Drive

To remove a mini-disc from your disc drive, follow this procedure (try it!):

1. Make sure the drive is turned on.
2. Lift the drive door and carefully pull the disc out.
3. Return the disc to its protective envelope and store it upright in a dust-free box.
4. Close the drive door.

Write-protecting 3½-inch Micro-discs

The micro-disc is write-enabled when you receive it. That means you can write data on the disc at any time. Any data on the disc when you do the writing operation is overwritten and therefore destroyed. If you want to ensure that the data on the disc is preserved (not overwritten), you can write-protect the disc. To prepare the disc for write-protected operation do the following:



1. Weaken or score the attach point so as not to break the tab.
   Break off the write-protect tab.

2. Align the protrusion on the tab with the groove in the disc.

3. Depress the tab into the groove–tab should fit snugly. In the position shown, the tab is write-protecting the disc.

Write-protect

4. To write-enable the disc, slide the tab up.

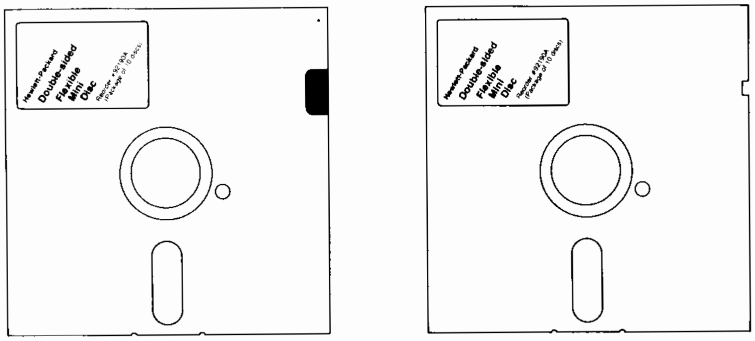
Write-enable

3½-inch Micro-disc Write-protection

Write-protecting 5¼-inch Mini-discs

Covering or uncovering a notch in the disc jacket determines whether the disc drive can write information on the disc. When the notch is covered, it's impossible for the drive to write on the disc; thus information already on the disc is protected from being written over or erased. This is useful when a disc contains source information which should only be read.

Labels are supplied with discs to allow you to cover the write-enable notch.



Write-protected **Write-enabled**

5¼-inch Mini-disc Write-protection

Booting BASIC

Normally, only one system program is accessible to your computer at a given time. When this is true, booting BASIC is a simple operation.

You may, however, have more than one system program that is accessible. For example, you may have an operating or language system residing on a hard disc drive connected to your computer. In that case, a different procedure is required to boot BASIC from flexible discs.

Both procedures are provided.

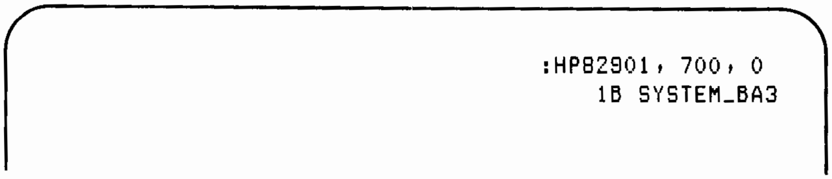
Booting with One System Present

Use this procedure if only one system program is accessible to your computer at any given time. This is the usual case. If you have multiple systems that your computer can access, move on to the next section, "Booting with Multiple Systems Present".

1. Turn your computer off.
2. Follow the previous disc-handling instructions and load your *System Disc* into a connected disc drive as follows:
   - If using a system on 5¼-inch (mini) flexible discs, insert the disc–label side up and media window forward–into your disc drive and close the door.
   - If using a system on 3½-inch (micro) flexible discs, insert the disc into your disc drive according to the instructions in the preceding section.
3. Turn your computer on. The computer runs its self-test, and then searches all devices for a system to boot. Since you have only one system "on line", this system will be found and booted. The message:

   ```
   BOOTING A SYSTEM
   ```

   will then be displayed. The name of the system being booted and the location where it was found are displayed in the upper-right portion of the screen. See the following example for a BASIC system found on the left drive of an HP 82901M disc drive connected to the internal HP-IB at bus address 0.



```
                                         :HP82901, 700, 0
                                            1B SYSTEM_BA3
```

Example of System Found

When booting is completed, the following message appears at the bottom of the display:

```
BASIC Ready 3.XX    (c) Copyright HP 1984
```

where XX is the current revision.

4. Turn the computer off and repeat step 3, except this time write down the top line of the message (`:HP82901, 700, 0` in the example) on a sticky label and affix the label to the flexible disc drive into which you inserted the *System Disc.* The top line of the message is the mass storage unit specifier (msus). The msus is explained in Chapter 6.
5. If you have other flexible disc drives in or connected to your computer, repeat steps 1 through 4 for each drive. When you're through, you should know the msus for every flexible disc drive, and every drive should be labeled with its msus.

Problems?

System Not Found. Your computer will look once at each internal or connected device for an operating system. If none is found, it displays the following message on the screen:

```
SEARCHING FOR A SYSTEM (ENTER To Pause)
```

The computer keeps cycling through its search routine until it finds a system. If the computer takes several seconds and still doesn't begin booting, start checking for problems. A few common ones are listed in the following table.

Problem	Solution
Drive containing system is not tumed on.	Turn drive on. Computer will automatically find system.
System Disc is not inserted in drive.	Insert disc and press reset key.
Drive is not connected to computer.	Turn computer off, connect drive, and tum computer on.
Drive door is not closed.	Close drive door.

Not Enough Memory. You must have enough RAM in your computer to hold your BASIC system. If the computer runs out of memory while loading the system, the following message is displayed:

```
NOT ENOUGH MEMORY
```

To solve this problem you must install another memory card as described in your computer's *Installation Guide.*

Booting with Multiple Systems Present

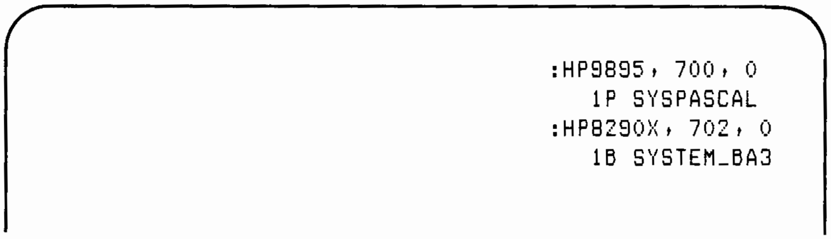
Note

The following procedure can only be done with boot ROMs of revision 3.0 and later.

The computer uses a searching sequence when looking for system programs and boots the first system it finds. If you have multiple system programs accessible to your computer, you will want to select the BASIC system to be booted.

1. Turn your computer off.
2. Follow the previous disc-handling instructions and load your *System Disc* into a connected disc drive as follows:
   - If using a system on 5¼-inch (mini) flexible discs, insert the disc–label side up and media window forward–into your disc drive and close the door.
   - If using a system on 3½-inch (micro) flexible discs, insert the disc into your disc drive according to the instructions in the preceding section.

3. Turn your computer on. While the computer is running its self-test, press the space bar a few times. This signals the computer that you want to override its default selection and choose which system to boot. This technique does not have an immediate effect, so don't be concerned if the computer doesn't seem to respond.
4. Review all systems found. The computer finds all systems it can and displays their names and where they were found in the upper right portion of the screen (see example).



```
:HP9895, 700, 0
   1P SYSPASCAL
:HP8290X, 702, 0
   1B SYSTEM_BA3
```

Example of Systems Found

5. Select your system. Beneath the name of the device where the system was found is a code followed by the name of the system file. This is the code you type to select the system you want. All systems except ROM-based systems consist of one or two numbers followed by a letter. The codes for ROMs are single letters. To choose a system, just type in its code. So, to boot BASIC in the example, type `1B`. The code you type is displayed in the lower right corner of the screen. The computer boots the system and displays the following message at the bottom of the screen:

   ```
   BOOTING A SYSTEM
   ```

   When booting is completed, the following message appears at the bottom of the display:

   ```
   BASIC Ready 3.XX    (c) Copyright HP 1984
   ```

   where XX is the current revision.
6. Turn the computer off and repeat step 3. When the systems list appears in the upper right corner of the display, write down the top line of the BASIC system message (`:HP8290X, 702, 0` in the example) on a sticky label. Affix the label to the flexible disc drive into which you inserted the *System Disc.* The top line of the BASIC system message is the mass storage unit specifier (msus). The msus is explained in Chapter 6.

7. If you have other flexible disc drives in or connected to your computer, repeat steps 1 through 6 for each drive. When you're through, you should know the msus for every flexible disc drive, and every drive should be labeled with its msus.

Problems?

System Not Found. Your computer will keep cycling through its search routine until it finds a system. If the BASIC system fails to appear in the list of available systems, something is preventing the computer from finding it. A few common causes of this problem are listed below:

Problem	Solution
Drive containing system is not turned on.	Turn drive on. Computer will automatically find system.
System Disc is not inserted in drive.	Insert disc and press reset key.
Drive is not connected to computer.	Turn computer off, connect drive, and turn computer on.
Drive door is not closed.	Close drive door.

Not Enough Memory. You must have enough RAM in your computer to hold your BASIC system. If the computer runs out of memory while loading the system, the following message is displayed:

```
NOT ENOUGH MEMORY
```

To solve this problem you must install another memory card as described in your computer's *Installation Guide.*

Loading BIN Files

When you've finished loading the BASIC system into your computer, indicated by the `BASIC Ready` message, you have the core BASIC language loaded into your computer. However, to "talk" with your peripherals and to do other useful work, you need to load BIN files from your *Drivers Disc* and/or the *Language Extensions Disc.* Chapter 8 contains detailed information on BIN files, but for now, continue here.

What are BIN Files?

There are two kinds of BIN files:

1. drivers (contained on the *Drivers Disc*), and
2. language extensions (contained on the *Language Extensions Disc*).

Drivers are simply programs that enable certain operations to occur. For example, the DISC and HPIB drivers must be loaded before you can `CAT` the contents of many external flexible discs (see what's on them). Language extensions are BASIC statements for specialized operations. For example, the BIN file GRAPH contains BASIC graphics statements that are not part of core BASIC.

Now Do This!

This procedure assumes that you have loaded the *System Disc* and the message `BASIC Ready` appears at the bottom of your display.

You will now load BIN file ERR into your computer by using the `LOAD BIN` command. The ERR BIN file provides a textual error message in addition to an error number when you tell the computer to do something it doesn't like or understand. This helps you to easily identify and correct the error. Without ERR loaded, only the error number is displayed.

You will also be directed to load certain BIN files depending on your system configuration.

1. Remove the *System Disc* from the flexible disc drive, and insert the *Language Extensions Disc* into the same drive.

Note

After typing in each line, you press a specific key on your keyboard to execute the `LOAD BIN` command. The key you press depends on the keyboard you have:

HP 46020A - Return
HP 98203B - EXECUTE
HP 98203A - EXEC

Every required key press in this guide lists all three keys if they differ. The keys are listed in the keyboard order shown above (e.g., Return, EXECUTE or EXEC).

If you have a difficult time using your keyboard, look over the section that describes your keyboard in Chapter 4. Then come back here to do the procedure.

Non-US ASCII Keyboards

This note applies only if you have a non-US ASCII keyboard (except Katakana); if you have a US ASCII or Katakana keyboard, skip to Step 2.

The non-US ASCII keyboard operates as a US ASCII keyboard until the LEX BIN file is loaded. The characters displayed correspond to U.S. keyboard positions without regard for the legend on the keycaps.

To configure your keyboard so that all characters displayed match the corresponding keycaps, you must load the LEX BIN file. Follow the procedure for your keyboard.

HP 46020 Keyboard
To display the "A" character with the French or Belgian keyboard, you must press (Q).

To display the """ character with all keyboards, you must hold down the key identified by ① in Figure 1, and press the key identified by ② .

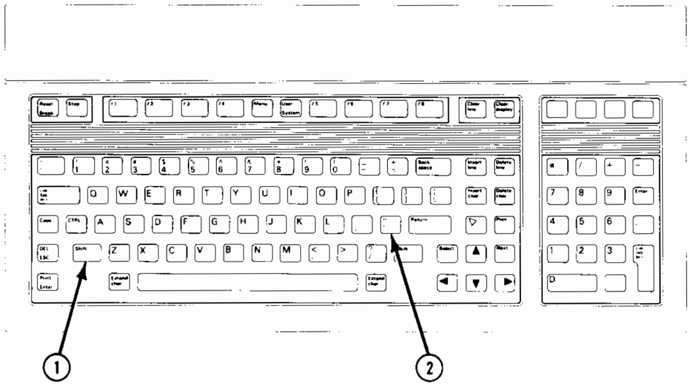


Figure 1

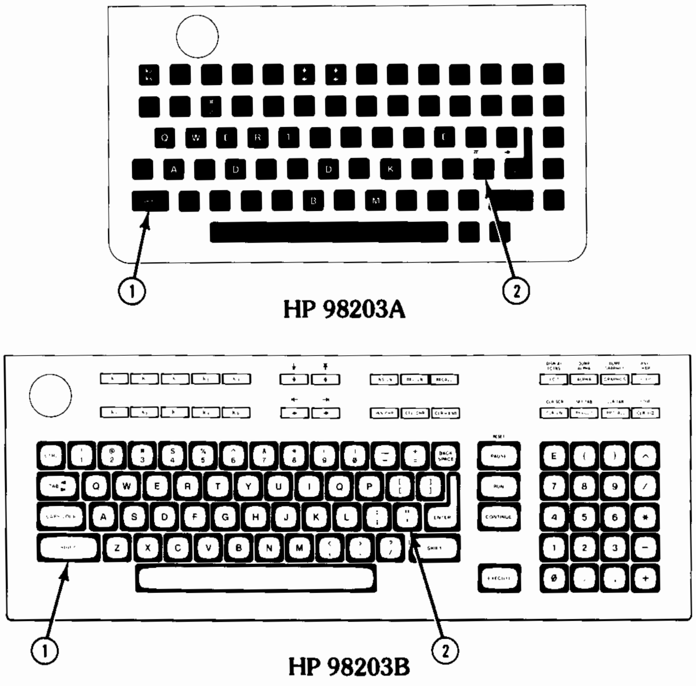
Type:

```
LOAD BIN "LEX" (Return)
```

When the message `BASIC LEX 3.0` is displayed, skip to Step 2. The characters displayed will now match the legends on your keycaps.

Non-US ASCII Keyboards, Continued

HP 98203A/B Keyboard
To display the ““"”” character, you must hold down the key identified by ① in Figure 2, and press the key identified by ②



HP 98203A

HP 98203B

Figure 2

Type:

`LOAD BIN "LEX"` EXECUTE or EXEC

When the message `BASIC LEX 3.0` appears, continue with Step 2. The characters displayed will now match the legends on your keycaps.

2. Type:

   `LOAD BIN "ERR"` Return, EXECUTE or EXEC

   After a few seconds the following message should appear at the bottom of the display, indicating that BIN file ERR has been loaded into the computer:

```
BASIC ERR 3.0    (c) Copyright HP 1984
```

3. **If** you used an internal disc drive (one built into your computer) to load the BASIC system and the ERR BIN file, remove the *Language Extensions Disc* from your drive and skip to "What's Next?" at the end of this chapter.

   Otherwise, remove the *Language Extensions Disc* from the flexible disc drive, insert the *Drivers Disc* into the same drive, and

   Type:

   `LOAD BIN "DISC"` **Return**, **EXECUTE** or **EXEC**

   After a few seconds the following message should appear at the bottom of the display, indicating that BIN file DISC has been loaded into the computer:

   ```
   BASIC DISC 3.0     (c) Copyright HP 1984
   ```

   Type:

   `LOAD BIN "CS80"` **Return**, **EXECUTE** or **EXEC**

   After a few seconds the following message should appear at the bottom of the display, indicating that BIN file CS80 has been loaded into the computer:

   ```
   BASIC CS80 3.0     (c) Copyright HP 1984
   ```

   Type:

   `LOAD BIN "HPIB"` **Return**, **EXECUTE** or **EXEC**

   After a few seconds the following message should appear at the bottom of the display, indicating that BIN file HPIB has been loaded into the computer:

   ```
   BASIC HPIB 3.0     (c) Copyright HP 1984
   ```

4. Remove the *Drivers Disc* from the drive.

What's Next?

You have loaded the BASIC system and BIN file ERR. If you loaded from an external disc drive, you also loaded BIN files DISC, CS80, and HPIB.

Go on to Chapter 3 where you will back up your BASIC discs.

Backing Up BASIC

Chapter 3

Flexible discs are an extremely reliable storage media. However, like phonograph records, they do wear out. Since discs can also be damaged due to accidents or careless handling, you should keep a duplicate or back-up copy of each important disc.

This chapter provides the information you need to back up your BASIC language system discs. First, you need to initialize new discs or used discs whose programs/data you no longer want. Then you will copy your BASIC language system discs to the initialized new or used discs. Finally, you will store your original BASIC discs in a safe place.

Initializing Flexible Discs

If you need a supply of flexible discs, order HP Part No. 9164-0175 for 5¼-inch mini-discs or HP Part No. 9164-0198 for 3½-inch micro-discs.

Before a disc can be used, it must be formatted or initialized for use with your computer. This process checks the disc for defects (areas where information cannot be stored) and then creates a file directory. The directory holds the name and location of each file on the disc. When a disc is first initialized, the directory is empty.

Some disc drives provide disc formatting options that are selected with the INITIALIZE statement. For example, you can select single-sided or double-sided when initializing a disc on the HP 9122 disc drive. Refer to your disc drive manual for details.

CAUTION

INITIALIZING A DISC DESTROYS DATA OR PROGRAMS RECORDED ON THE MEDIA. BE SURE YOU KNOW WHAT DISC YOU ARE USING. YOU CAN INADVERTENTLY WIPE OUT YOUR BASIC 3.0 LANGUAGE SYSTEM, OR OTHER FILES, IF YOU INITIALIZE THE WRONG DISC.

IF YOU INADVERTENTLY INITIALIZE A HARD DISC DRIVE, DO NOT INTERRUPT THE INITIALIZATION PROCESS. WAIT UNTIL THE DISC DRIVE LIGHT GOES OFF BEFORE CONTINUING. THIS CAN TAKE OVER AN HOUR. INTERRUPTING A HARD DISC DRIVE INITIALIZATION CAN SEVERELY DAMAGE THE DRIVE.

To initialize a new disc, or re-initialize (erase all files) on an old disc:

1. Ensure that BASIC and the required BIN files are loaded (Chapter 2), and that you have removed any disc from your flexible disc drive.
2. Insert the new or used disc into a flexible disc drive. Be sure the disc is write-enabled (Chapter 2).
3. Type:

   ```
   INITIALIZE ":msus"
   ```

   where: **msus** is the mass storage unit specifier defined when BASIC was booted from the disc drive you're now using

   Examples: `INITIALIZE ":HP8290X,702,0"`
   `INITIALIZE ":INTERNAL"`
4. Press: (Return), (ENTER) or (ENTER)

   The initialization routine runs automatically. An asterisk (*) appears in the lower right corner of the display, and the disc drive light turns on to indicate initialization is occurring. (If you're initializing on an internal disc drive, the Model 226 or 236 displays a message indicating the disc is being initialized.) Initialization takes several minutes. When the disc is initialized, the disc drive light goes off and the asterisk goes away.
5. Remove the initialized disc from the disc drive. Repeat the process for each disc you want to initialize. You will need six discs to make one copy of each of your six BASIC language system discs.

Copying Your BASIC Discs

The disc copying procedure depends on how many flexible disc drives are in your system. If you have two or more drives that you can use, continue here. If you have only one flexible disc drive, skip to the next procedure.

Copying Discs With Two Drives

1. Insert one of your original BASIC discs into a flexible disc drive.
2. Insert an initialized blank disc into another flexible disc drive.
3. Type:

   `COPY ":`**source msus**`" TO ":`**destination msus**`"`

   where: the **source msus** is the mass storage unit specifier of the drive containing the BASIC System disc, and **destination msus** is the mass storage unit specifier of the drive containing the blank disc

   Examples: `COPY ":INTERNAL,4,0" TO ":INTERNAL,4,1"` or
   `COPY ":4,0" TO ":4,1"`

   `COPY ":HP82901,700,0" TO ":HP82901,700,1"`
4. Press: (Return), (ENTER) or (ENTER)

   The asterisk (*) lights at the lower right of the display, and the lights on the disc drives go on to indicate that the copy operation is in progress. Copying takes about one minute. When it's done, the asterisk goes away and the disc drive lights go off.
5. Remove the original BASIC disc from its drive and the duplicate disc from its drive.
6. Label the duplicate disc with the same information on the original disc. Write "Copy" and the date on the label.
7. Repeat steps 1 through 6 for each original BASIC disc until you have a complete set of duplicate discs.

Copying Discs With One Drive

Note

This procedure may not work if your computer has limited memory. Refer to the CBACKUP description in the *BASIC 3.0 Utilities Library* for more information if this procedure doesn't work.

1. Type:

   ```
   INITIALIZE ":MEMORY,0"
   ```

   Press: Return, ENTER or ENTER

2. Insert one of your original BASIC discs into the flexible disc drive.
3. Type:

   ```
   COPY ":msus" TO ":MEMORY,0"
   ```

   where: **msus** is the mass storage unit specifier of the flexible disc drive

   Press: Return, ENTER or ENTER

   The asterisk (*) lights at the lower right of the display, and the lights on the disc drives go on to indicate that the copy operation is in progress. Copying takes several minutes. When it's done, the asterisk goes away and the disc drive lights go off.

4. When copying is done, remove the BASIC disc from the drive and insert an initialized blank disc into the drive.
5. Type:

   ```
   COPY ":MEMORY,0" TO "msus"
   ```

   where: **msus** is the mass storage unit specifier of the flexible disc drive

   Press: Return, ENTER or ENTER

   The asterisk (*) lights at the lower right of the display, and the lights on the disc drives go on to indicate that the copy operation is in progress. Copying takes about one minute. When it's done, the asterisk goes away, and the disc drive lights go off.

6. Remove the duplicate disc from the drive and label the disc with the same information on the original BASIC disc. Write "Copy" and the date on the label.
7. Repeat steps 2 through 6 for each original BASIC disc until you have a complet set of duplicate discs.

Storing Original Discs

You now have a set of original BASIC discs and a set of duplicate discs. Keep your duplicate discs with the computer for use as needed. Store your original discs in a secure place away from the computer. If your duplicate discs are lost, damaged, or wear out, use your original discs to boot BASIC and create new duplicate discs.

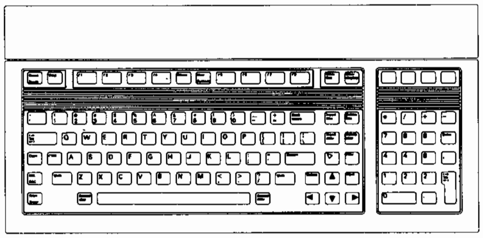
You may want to keep the original discs in a fireproof, waterproof storage area in a separate building if the data is critical.

Using Your Keyboard With BASIC

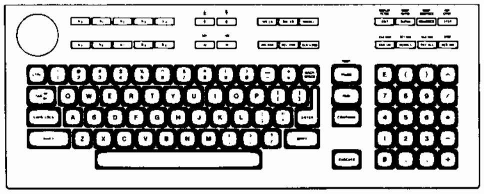
Chapter 4

Remember that the BASIC language system is a *system* program: it defines the keyboard, organizes the display, handles discussions between the processor and the peripherals–*plus* supports the BASIC language. In this chapter, we'll explain how the display is organized and help you become familiar with your keyboard.

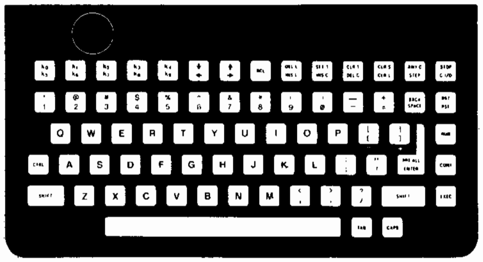
The BASIC system supports three keyboards:



HP 46020A



HP 98203B



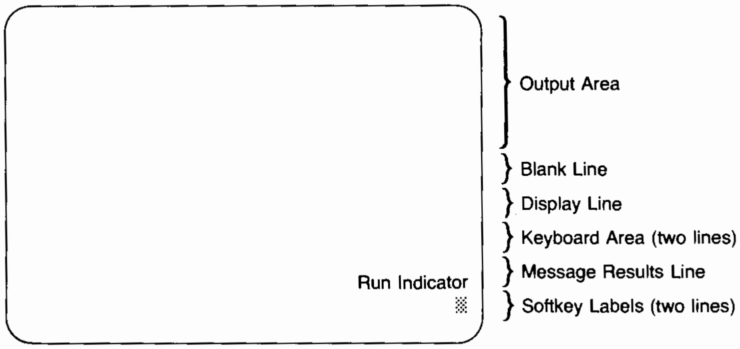
HP 98203A

After describing display organization, this chapter describes each of the keyboards and its key functions. The keyboards are described in the order in which they're shown in the preceding illustrations. Find the section that describes your keyboard and work through that section.

The exercises in this chapter assume you have already booted BASIC and loaded any required BIN files as described in Chapter 3.

The BASIC Display Organization

The BASIC language system partitions the screen into the following areas:

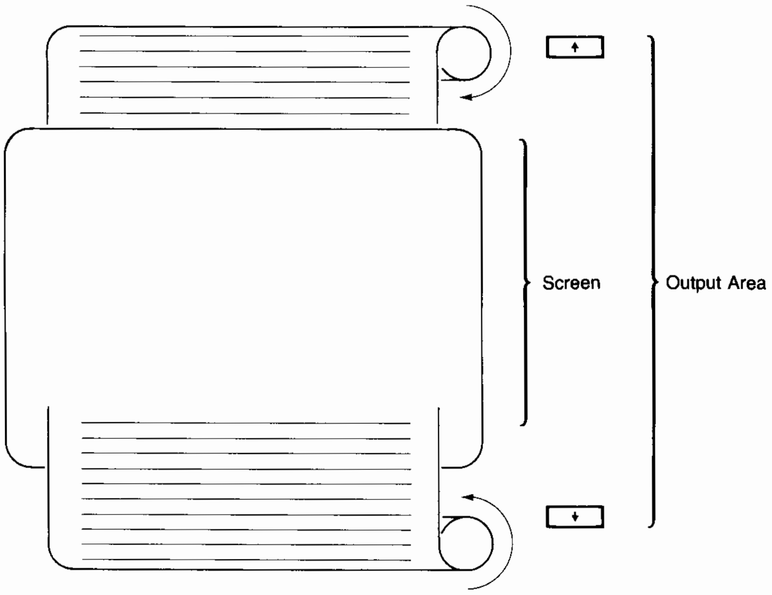


BASIC Display Organization

There are several types of displays that are supported by the BASIC language system. Displays differ in two major ways:

- Number of total lines that are displayed
- Number of characters per line

The **output area** can hold more lines of information than can appear on the screen at any one time. Think of the output area as a large multi-line scroll that rolls up and down in front of a small multi-line window. To view lines that you can't presently see, you simply "scroll" a new set of lines in front of the window. This "scrolling" effect is accomplished with the arrow keys, the cursor control wheel, or the "mouse" (HP 46060A). When the entire output area is filled, each new line entered causes the top-most line to be permanently lost.



The Output Area

Running programs use the **display line** to communicate with the operator.

The **keyboard area** echoes whatever you type at the keyboard, up to two lines.

The **message/results line** displays answers to arithmetic operations typed at the keyboard, as well as error messages from the BASIC language system.

Try It Yourself

Note

Don't worry about making a mistake. You won't hurt anything. The computer beeps and gives an error message to help you understand what you did wrong. Check your typing, which must be precise.

Note

When three keys are shown together
(e.g., Clear display, CLR SCR or CLR S),
the first key is for the HP 46020A, the second for the HP 98203B, and the third for the HP 98203A. Press the key specified for your keyboard.

First press Clear display, CLR SCR, or CLR S. Now try this arithmetic problem:

`98+26` Return, EXECUTE, or EXEC

Note

Whenever you see a key symbol
(e.g., Return, EXECUTE or EXEC)
following a line, press that key after typing the line.

The operation is first displayed in the keyboard area. When executed, the result `124` appears in the message/results line. To generate an error message in the message/results line, type:

`98/0` Return, EXECUTE, or EXEC

The following message is displayed:

```
ERROR 31  Division by 0 or X MOD 0
```

The **softkey labels** are defined on booting BASIC with the HP 46020A. With the other keyboards, loading BIN file KBD defines softkeys.

The **run indicator** shows what the computer is doing at any given time. When the indicator is blank, the computer is free and awaiting your command. Run indicators are given in the following table:

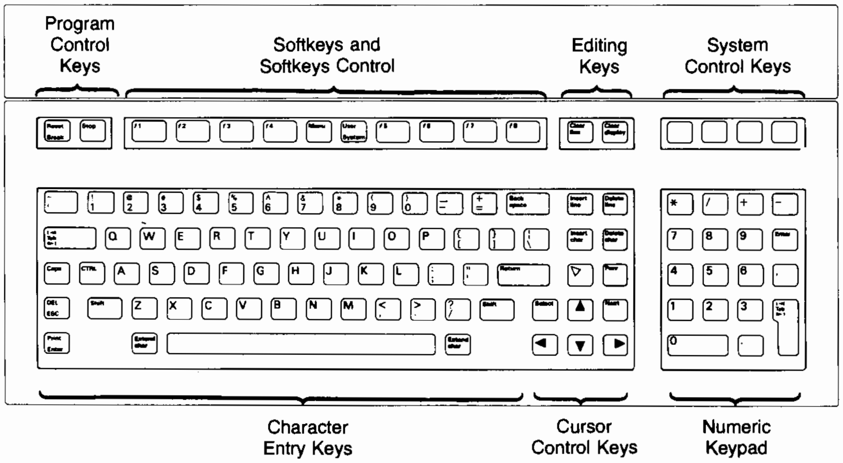
Symbol	Meaning
▒	Running a program.
–	Paused in a program.
blank	Stopped.
?	Waiting for a keyboard input.
*	Keyboard execution.
IO	Paused; waiting for an I/O operation to complete.

HP 46020A Keyboard Definitions

Note

If you do not have the HP 46020A keyboard, skip ahead to one of the following sections, which describe the HP 98203B and HP 98203A keyboards.

The keys on the HP 46020A keyboard are arranged into the following functional groups.



HP 46020A Keyboard

This section provides a handy reference guide to BASIC's key definitions for the HP 46020A keyboard. Keep in mind that other system programs may define the keys differently. Each key will be demonstrated where possible. One point to clarify: the **cursor** that we refer to in the following paragraphs is the blinking-underline that points to a location on the screen. (If you have a Model 237 computer, the cursor does not blink.)

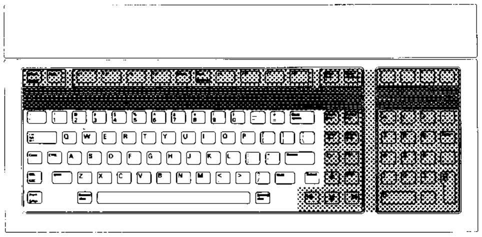
The optional HP 46060A mouse plugs into the back of the keyboard. The mouse allows you to quickly move the cursor in all directions and to scroll the display. It also has programmer-definable buttons.

Note

Before you proceed, type:

```
    SCRATCH (Return)
```

This clears the computer of any programs that might be left in memory from previous demonstrations.



Character Entry Keys

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

The Caps key sets the unshifted keyboard to either uppercase (which is the default after BASIC is booted) or lowercase (normal typewriter operation). The computer displays which mode the computer is in when you press the Caps key.

Type a few words, then press Caps and continue typing. Notice the case change. Press Shift-Clear line when finished.

You can enter standard uppercase and lowercase letters, using the Shift key to access the alternate case.

Type a few words, pressing Shift to change the case of the first letter of each word. Now press Caps and continue typing. Notice that the alternate case accessed by Shift depends on the setting of Caps. Press Shift-Clear line when finished.

The Return key has three functions:

1. When a running program prompts you for data, you respond by typing the requested data and then pressing Return. This signals the program that you have provided the data and that it can resume execution.
2. When typing in lines of a program, the Return key is used to store each line of program code.
3. After typing in a command, the Return key causes the command to be executed.

Type `EDIT` and press (Return). Notice the number `10` now displayed on the screen–this is the line number of the first line of a BASIC program. The computer is waiting for you to type in the line. Type:

```
!FIRST LINE
```

and press (Return). Notice that the computer accepts the statement as a program line and displays `20` in preparation for the next one. Press (Stop) when finished.

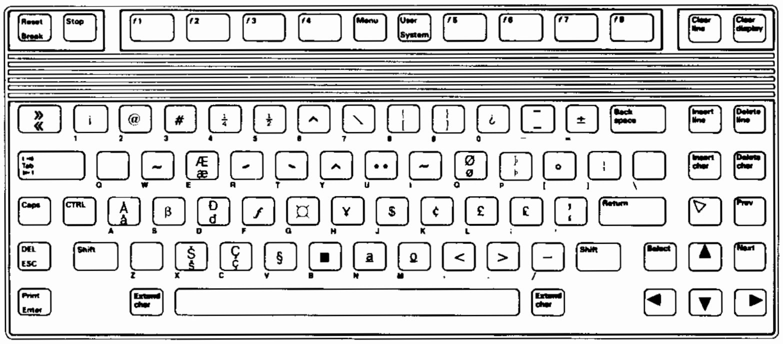
(Print Enter)

Pressing (Enter) is the same as pressing the (Return) key.

Pressing (Print) prints a complete copy of the alpha display on the default printer. The shifted version of the key directly above the (/) key in the numeric keypad performs the same function.

(Extend char)

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 (see illustration). On a Katakana keyboard, the "Roman" and "Katakana" keys select the other character sets.



Extended Character Set

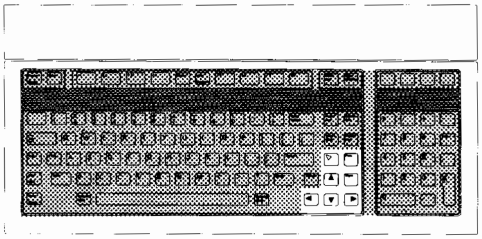
The Tab key moves the cursor forward to preset tabs. Pressing Shift-Tab moves the cursor back to preset tabs.

Before Tab can be used, a tab must be set. Tabs are set and cleared with System menu softkeys. The Tab key is demonstrated along with the **Set Tab/Clr Tab** softkey under "System Softkeys" later in this section.

The CTRL (control) key works like Shift to access a set of standard control characters, such as line-feed and form-feed. These characters are useful to the programmer for controlling some devices and for communicating with other computers. You probably won't need them when running programs. The available control characters are listed in the *BASIC 3.0 Language Reference*, Useful Tables section.

The Select key beeps but performs no function unless it is program-defined.

Cursor-Control Keys



The cursor-control keys move the display cursor. 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. The ▶ and ◀ keys allow you 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.

The unshifted ▷ key positions the print position at the beginning position on the page. The shifted ▷ 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 ▷ key, press Clear display. Then type `PRINT "SOMETHING"` and press Return; repeat twice. You should now have the following display:

```
SOMETHING
SOMETHING
SOMETHING
```

Press the ▷ key (unshifted).

Type `PRINT "ANY "` and press Return. Your display should look like this:

```
ANY THING
SOMETHING
SOMETHING
```

Press Clear display.

In normal mode, pressing the Prev key causes the display to scroll down one page and pressing the Next key causes the display to scroll up one page. In edit mode, these keys move the display one-half page.

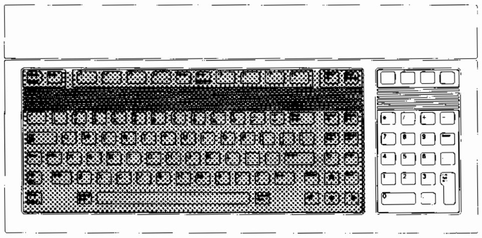
To test the horizontal movement of the cursor, type a few words and press the shifted and unshifted (◀) and (▶) keys. Notice that the cursor cannot be moved beyond the characters you have typed. Press (Shift)-(Clear line) when finished.

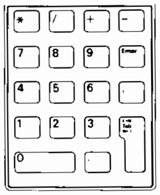
To test the vertical movement of the cursor, type `EDIT` and press (Return). Now type the following lines, pressing (Return) after each line (the first line may be there already, so just press (Return) to accept it):

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

Try out the shifted and unshifted (▲), (▼), and (▽) keys. Then try the (Prev) and (Next) keys. When you're done, press (Stop) to exit. Then, type `SCRATCH` (Return) 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, then press (Enter). The result is displayed in the lower-left corner of the screen.

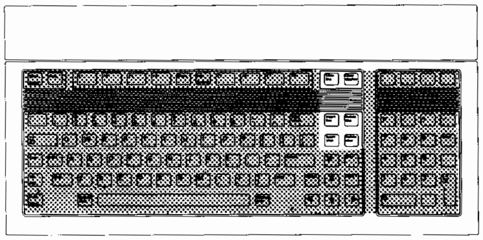
The (Enter) key performs the same function as the (Return) key. The (Tab) key on the numeric pad functions like the (Tab) key in the character entry area. The shifted versions of the (*), (/), (+), and (-) keys are E, (,), and ^, respectively. The shifted versions are also available in the character entry area.

Type in the following problem using the numeric pad:

```
(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.

Insert line

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

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

```
10 !FIRST LINE
```

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

Delete line

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

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

```
10 !FIRST LINE
```

and press (Delete line). The line is removed. To restore it, press the key directly above (↓) to recall it, then press (Return) to enter it into the program. Press (Stop) to exit edit mode.

Insert char

Pressing (Insert char) 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 char. Now type:

```
OF INSERT MODE
```

and press Insert char 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-Clear line when finished.

Delete char

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

Type a few words and experiment with Delete char, 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.

Clear line

Pressing unshifted-Clear line clears from the current cursor position to the end of the line.

Pressing Shift-Clear line clears the keyboard line and message/results line.

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

Clear display

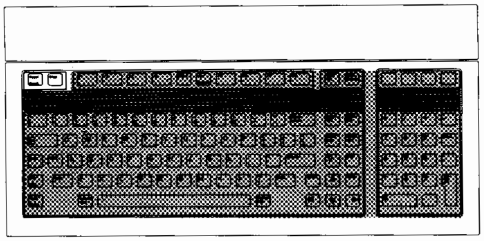
Pressing either the shifted or unshifted version of Clear display clears the entire alpha screen.

Type the following BASIC command:

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

Now press Return to execute it. Press the key directly above * to recall the command, and press Return again. Repeat this step several times to fill the screen with messages. Now press Clear display 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.

Stop

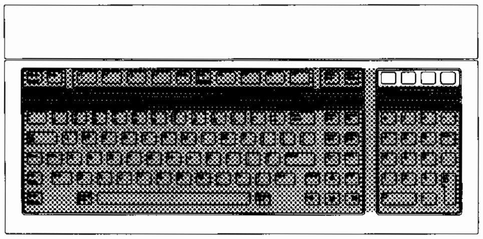
Pressing unshifted-(Stop) **pauses** program execution after the current line. Pressing **Continue** (unshifted (f2)) with System menu active resumes program execution from the point where it was paused.

Pressing (Shift)-(Stop) stops program execution after the current line. To restart the program, press **RUN** (unshifted (f3)) with System menu active.

Reset
Break

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

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



System Control Keys

Four unlabeled keys directly above the numeric keypad control various system functions related to the display, printer, and editing operations. Most of these keys execute their functions immediately, as the key is pressed.

To easily identify the keys in the following description, we'll use this convention:

- Key 1 - Above the (*) key.
- Key 2 - Above the (/) key.
- Key 3 - Above the (+) key.
- Key 4 - Above the (-) key.

Key 1 - Recall
Pressing unshifted-Key 1 (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"  (Return)
```

to print the number 1 on the screen. Now press Key 1 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 (Return) again. Now press Key 1 several times to see all of the statements it remembers. Then press (Clear display) when finished.

(Shift)-Key 1 moves forward through the recall stack.

Pressing (f8) with System menu active performs the same recall function as Key 1.

Key 2 - Alpha/Dump Alpha

Pressing unshifted-Key 2 (Alpha) once turns on the alphanumeric display. Pressing it the second time turns off the graphics display. This key function requires that the GRAPH BIN file be loaded. If you have a Model 237, this key performs no function.

Pressing (Shift)-Key 2 (Dump Alpha) prints a complete copy of the alpha display on the default printer. The Dump Alpha function is also executed by (Print).

Key 3 - Graphics/Dump Graphics

Pressing unshifted-Key 3 (Graphics) once turns on the graphics display. Pressing it the second time turns off the alphanumeric display. If you have a Model 237, this key performs no function.

Pressing (Shift)-Key 3 (Dump Graphics) prints a complete copy of the graphics display on the default printer. If you have a Model 237, the combined alpha and graphics display is printed.

Both key functions require that the GRAPH BIN file be loaded.

Key 4 - Result

Pressing Key 4 (Result) either shifted or unshifted returns the result of the last arithmetic expression that was executed.

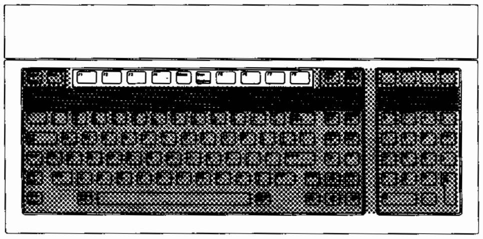
Press (Shift)-(Clear line), then type:

```
23+45 (Return)
```

The result, `68`, is displayed in the lower left corner of the screen. To add 123 to this value, press Key 4 and type:

```
+123 (Return)
```

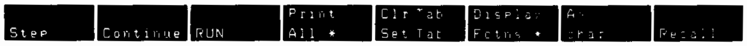
The new result, `191`, is now displayed. Press (Shift)-(Clear line) when finished.



Softkeys and Softkeys Control

There are eight softkeys (labeled f1 through f8) and two keys that control the definitions of the softkeys (Menu and System).

When the BASIC system is booted, the softkeys default to System mode. The System mode menu that appears at the bottom of your display is shown. 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. *BASIC 3.0 Programming Techniques*, Chapter 2, describes how to set up User modes.



Softkeys Control Keys

User
System

Pressing unshifted-System causes softkeys to assume System mode. The System menu is displayed, **if** the Menu key is toggled to the "on" position.

Pressing Shift-User puts the softkeys in User 1 mode. The User 1 menu is displayed **if** the Menu key is toggled to the "on" position.

Menu

Pressing unshifted-Menu toggles the softkey labels— turns them on if they're off and turns them off if they're on.

Pressing Shift-Menu increments User mode and menu **if** User mode is "on".

User menus are blank unless BIN file KBD is loaded.

Try It Yourself

Let's get familiar with the two 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 System. If it is still not displayed, press Menu.

With the System menu displayed, press unshifted-Menu several times. The System menu display should go on and off. Leave the System menu displayed, and continue.

Now press Shift-User. The User 1 menu should appear on your display.

Press 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 unshifted-Menu several times and the last User menu goes on and off. Leave the User menu on.

Finish this exercise by pressing unshifted-System to get your computer back in System mode.

System Softkeys

The following paragraphs define the eight System softkeys.

Step (unshifted-*f1*)

Step allows you to execute one program line at a time. This is particularly useful for debugging (fixing) programs.

Continue (unshifted-*f2*)

Continue resumes program execution from the point where it was paused (by an unshifted-Stop).

RUN (unshifted-*f3*)

RUN starts a program running from the beginning.

Print All (unshifted-f4)

The **Print All** key turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device. Press Print once to set printall "on" and again to set printall "off". An asterisk (*) appears next to `All` to indicate that printall is "on".

The display's output area is the default printall device at powerup. The *BASIC 3.0 Programming Techniques* explains how to select other printall devices.

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

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

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 Return
```

Because this is not an executable BASIC statement, an error message is displayed, both at the bottom of the screen and in the printall area at the top. This way, a log is produced of all commands typed and executed at the keyboard, along with any error messages. Press Clear display to clear the display, and press **Print All** to turn off printall mode.

Set Tab/Clr Tab (f5)

Set Tab (unshifted-f5) sets a tab at the cursor's current position. Tabs remain in effect until cleared by either **Clr Tab** or the SCRATCH A statement (explained in *BASIC 3.0 Programming Techniques*), Chapter 2.

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 **Set Tab**. Move the cursor back several spaces using ◀, then press 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 **Clr Tab**. Press Shift-Clear line when finished.

Display Fctns (unshifted-f6)

Display Fctns sets the display-functions mode, allowing you to see special control characters (e.g., form-feed, carriage return) on the screen. Pressing this key a second time cancels the display-functions mode. An asterisk (*) appears next to `Fctns` to indicate that display-functions mode is "on".

Type the following line:

```
PRINT "DISPLAY-FUNCTIONS MODE OFF"  Return
```

Notice the display at the top of the screen. Now press **Recall** (unshifted-f8) to recall the line, and edit it to read:

```
PRINT "DISPLAY-FUNCTIONS MODE ON"
```

Press **Display Fctns**, and then press Return. Notice that the carriage return (CR) and line-feed (LF) control characters are now displayed. Press **Display Fctns** again to exit display-functions mode. Press Clear display when finished.

Any char (unshifted-f7)

Any char is used to find any ASCII character. First press **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 section of the *BASIC 3.0 Language Reference.*

Press **Any char**, then type 189 which is the decimal equivalent of "§". The display line now displays "§". Press Shift-Clear line to erase it.

Recall (f8)

The **Recall** softkey (unshifted-f8) acts just like System Control Key 1 (described earlier). **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"   (Return)
```

to print the number 1 on the screen. Now press **Recall** 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 (Return) again. Now press **Recall** several times to see all of the statements it remembers. Note that **Recall** goes backward through the queue.

Pressing (Shift)-(*f8*) allows you to cycle forward through the queue until the last line entered, executed, or deleted is displayed. In the previous exercise you pressed unshifted-(*f8*) several times, cycling backward through the queue. Now press (Shift)-(*f8*) several times to cycle forward through the queue until the last line is displayed.

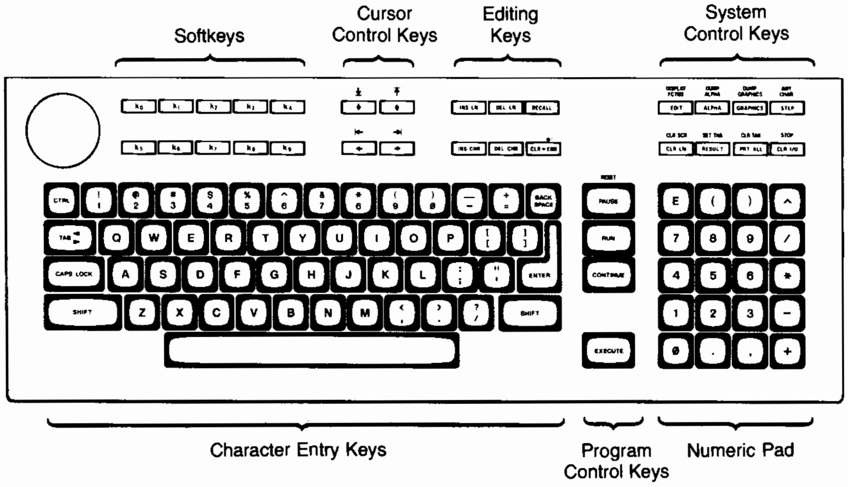
Now skip ahead to Chapter 5.

HP 98203B Keyboard Definitions

Note

If you have the HP 46020A keyboard, refer to the preceding section. If you have the HP 98203A keyboard, skip to the following section.

The keys on the HP 98203B keyboard are arranged into the following functional groups:



HP 98203B Keyboard

This section provides a handy reference guide to BASIC's key definitions for the HP 98203B keyboard. Keep in mind that other system programs may define the keys differently. Each key will be demonstrated where possible. One point to clarify: the **cursor** that we refer to in the following paragraphs is the blinking-underline that points to a location on the screen.

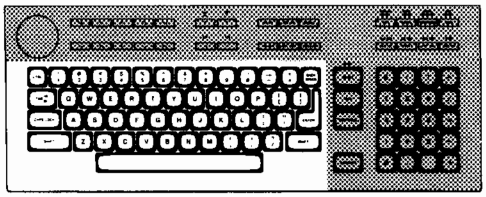
Note

Before you proceed, type:

```
    SCRATCH EXECUTE
```

This clears the computer of any programs that might be left in memory from previous demonstrations.

Character Entry Keys



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

The CAPS LOCK key sets the unshifted keyboard to either upper case (which is the default after BASIC is booted) or lower case (normal typewriter operation). The computer displays which mode the computer is in when you press the CAPS LOCK key.

Type a few words, then press CAPS LOCK and continue typing. Notice the case change. Press CLR LN when finished.

You can enter standard upper-case and lower-case letters, using the SHIFT key to access the alternate case.

Type a few words, pressing SHIFT to change the case of the first letter of each word. Now press CAPS LOCK and continue typing. Notice that the alternate case accessed by SHIFT depends on the setting of CAPS LOCK. Press CLR LN when finished.

The ENTER key has several functions:

1. When a running program prompts you for data, you respond by typing the requested data and then pressing ENTER. This signals the program that you have provided the data and that it can resume execution. The EXECUTE key can also be used for this function.

2. When typing in lines of a program, the ENTER key is used to store each line of program code. The EXECUTE key can also be used for this function.
3. Like the EXECUTE key, the ENTER key can be used to execute commands and calculations.

Type `EDIT` and press ENTER. Notice the number `10` now displayed on the screen–this is the line number of the first line of a BASIC program. The computer is waiting for you to type in the line. Type:

```
!FIRST LINE
```

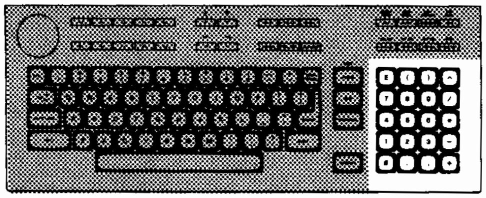
and press ENTER. Notice that the computer accepts the statement as a program line and displays `20` in preparation for the next one. Press PAUSE when finished.

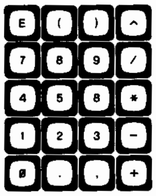
The TAB key moves the cursor forward to preset tabs. Pressing SHIFT-TAB moves the cursor back to preset tabs.

Before TAB can be used, a tab must be set. Press the space bar to move the cursor forward a few spaces and press SET TAB (SHIFT-RESULT). Move the cursor back several spaces using ←, then press 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 CLR TAB (SHIFT-PRT ALL). Press CLR LN when finished.

The CTRL (control) key works like SHIFT to access a set of standard control characters, such as line-feed and form-feed. These characters are useful to the programmer for controlling some devices and for communicating with other computers. You probably won't need them when running programs. The available control characters are listed in the Useful Tables section of *BASIC 3.0 Language Reference.*

Numeric Pad



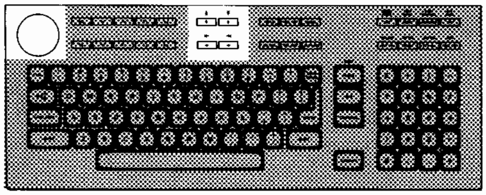


The numeric pad provides a convenient way to enter numbers and perform arithmetic operations. Simply type in the arithmetic expression you want to evaluate, then press (EXECUTE). The result is displayed in the lower-left corner of the screen.

Type in the following problem using the numeric pad:

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

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



Cursor-Control Keys

The cursor-control keys move the display cursor. The ↑ and ↓ keys allow you to scroll lines in the output area up and down. The → and ← keys allow you to move horizontally along a line. The **BACK SPACE** key works just like the ← key.

The cursor control wheel (also called the knob) allows you to rapidly scroll the print area (with **SHIFT** depressed) or move the cursor left and right (unshifted).

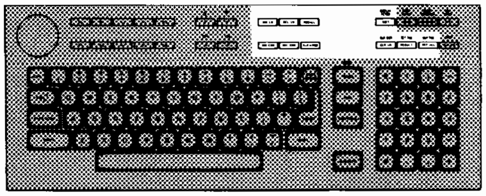
To test the horizontal movement of the cursor, type a few words and press the ← and → keys. Notice that the cursor cannot be moved beyond the characters you have typed. Now rotate the wheel to move the cursor. Press **CLR LN** when finished.

To test vertical scrolling, type `EDIT` and press **EXECUTE**. 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
```

Press the **SHIFT** key and rotate the wheel to scroll the text up and down. Also try out the ↑ and ↓ keys. When you're done, press **PAUSE** to exit.

Editing Keys



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

DISPLAY FCTNS
EDIT

The EDIT key is a typing convenience; pressing EDIT followed by EXECUTE puts the computer in program edit mode. Edit mode allows the programmer to enter and edit program lines.

Press EDIT, then EXECUTE to enter edit mode. The number 10 appears on the screen. This is a line number for a BASIC program; the computer is waiting for you to type in a line of code. If there is a program already in memory, the computer displays it on the screen. Press PAUSE to exit edit mode.

RECALL

The RECALL key 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" EXECUTE
```

to print the number 1 on the screen. Now press RECALL 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 EXECUTE again. Now press RECALL several times to see all of the statements it remembers. Then press CLR SCR when finished.

SHIFT-RECALL moves forward through the recall stack.

INS LN

INS LN inserts a new line above the cursor's current position (edit mode only).

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

```
10 !FIRST LINE
```

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

DEL LN

DEL LN deletes the line containing the cursor (edit mode only).

Press EDIT, then EXECUTE. Position the cursor to the line:

```
10 !FIRST LINE
```

and press DEL LN. The line is removed. To restore it, press RECALL to retrieve it, then ENTER to enter it into the program. Press PAUSE to exit edit mode.

INS CHR

INS CHR 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 INS CHR. Now type:

```
OF INSERT MODE
```

and press INS CHR 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 CLR LN when finished.

DEL CHR

DEL CHR deletes the character at the cursor's position.

Type a few words and experiment with DEL CHR, 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.

SET TAB
shift- RESULT

SET TAB sets a tab at the cursor's current position. Tabs remain in effect until cleared by either CLR TAB or the SCRATCH A statement. The SCRATCH commands are explained in *BASIC 3.0 Programming Techniques*, Chapter 2. To demonstrate SET TAB, see TAB.

CLR TAB
shift- PRT ALL

CLR TAB clears a tab previously set at the cursor's position. To demonstrate CLR TAB, see TAB.

CLR LN

CLR LN clears the keyboard line and message/results line.

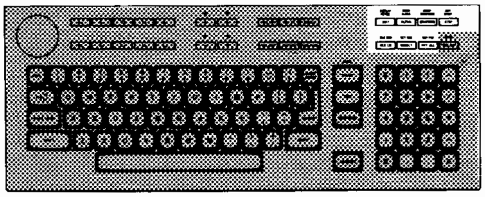
Type a few words and press CLR LN to clear them.

CLR→END

CLR→END clears from the current cursor position to the end of the line.

Type in a few words and use the cursor control wheel or ← to position the cursor in the middle of the line. Press CLR→END to clear to the end of the line. Press CLR LN to clear the rest of the line.

System Control Keys



These keys control various system functions related to the display, printer and editing operations. Most of these keys execute their functions immediately, as the key is pressed.

EDIT

EDIT types the EDIT command on the keyboard line. See the Editing Keys section for more information.

DISPLAY FCTNS
shift- EDIT

DISPLAY FCTNS sets the display- functions mode, allowing you to see special control characters (e.g., formfeed, carriage return) on the screen. Pressing this key a second time cancels the display-functions mode.

Type the following line:

```
PRINT "DISPLAY-FUNCTIONS MODE OFF"   EXECUTE
```

Notice the display at the top of the screen. Now press RECALL to recall the line, and edit it to read:

```
PRINT "DISPLAY-FUNCTIONS MODE ON"
```

Press the DISPLAY FCTNS key, and then press EXECUTE. Notice that the carriage return (CR) and line-feed (LF) control characters are now displayed. Press DISPLAY FCTNS again to exit display-functions mode. Press CLR SCR when finished.

ALPHA

GRAPHICS

ALPHA and GRAPHICS allow you to turn the alpha and graphics display modes on and off. These keys are demonstrated in Chapter 7. The GRAPH BIN file must be loaded for these keys to function.

DUMP ALPHA
shift- ALPHA

The DUMP ALPHA key prints a complete copy of the alpha display on the default printer. Refer to the "Dumping to a Printer" section of Chapter 6 for additional information.

DUMP GRAPHICS
shift- GRAPHICS

The DUMP GRAPHICS key prints a complete copy of the graphics display on the default printer. Refer to the "Dumping to a Printer" section of Chapter 6 for additional information. The GRAPH BIN file must be loaded for this key to function.

STEP

STEP allows the programmer to step through a program, one line at a time. Using the STEP key to debug programs is covered in *BASIC 3.0 Programming Techniques*, Chapter 12.

ANY CHAR
shift- STEP

ANY CHAR is used to find any ASCII character. First press ANY CHAR. Then enter a three-digit number from 000 thru 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 section of the *BASIC 3.0 Language Reference.*

Press ANY CHAR, then type 189. 189 is the decimal equivalent of "§", which is now displayed in the keyboard line. Press CLR LN to erase it.

CLR SCR
shift- CLR LN

CLR SCR clears the entire alpha screen.

Type the following BASIC command:

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

Now press EXECUTE to execute it. Press RECALL to recall the command and press EXECUTE again. Repeat this step several times to fill the screen with messages. Now press CLR SCR to erase all lines at once.

RESULT

RESULT returns the result of the last arithmetic expression that was executed.

Press CLR LN, then type:

```
23+45 EXECUTE
```

The result, 68, is displayed in the lower-left corner of the screen. To add 123 to this value, type:

```
RESULT +123 EXECUTE
```

The new result, 191, is now displayed. Press CLR LN when finished.

PRT ALL

The PRT ALL key turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device. Press PRT ALL once to set printall to ON and again to set printall to OFF. The printall mode is displayed in the lower-left corner of the screen.

The screen's output area is the default printall device at powerup. Selecting an external printall device is explained in *BASIC 3.0 Programming Techniques*, Chapter 8.

Press PRT ALL to turn on printall mode. Now type in the following command:

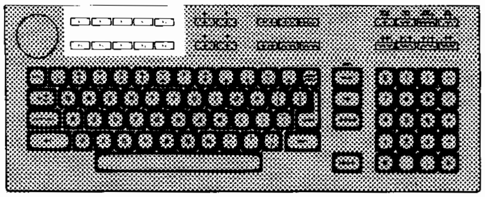
```
PRINT "THIS IS A KEYBOARD OPERATION" EXECUTE
```

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 EXECUTE
```

Because this is not an executable BASIC statement, an error message is displayed, both at the bottom of the screen and in the printall area at the top. This way, a log is produced of all commands typed and executed at the keyboard, along with any error messages. Press CLR SCR to clear the screen, and press PRT ALL to turn off printall mode.

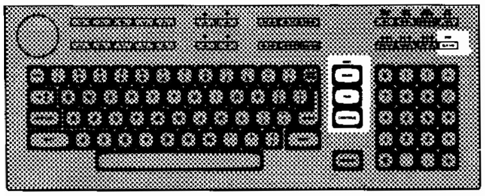
Softkeys



The ten keys labeled (k0) thru (k9) are defined under program control. The program may also display a label for each defined key. Pressing a defined key tells the computer to interrupt whatever it's doing and start running another part of the program.

We call these keys "softkeys" because the program or "software" defines and labels them. Another ten softkeys (without the displayed labels) can be defined at the same time and accessed with the (SHIFT) key. These shifted softkeys are often referred to as **k10** thru **k19**.

With BIN file KBD loaded, softkeys are defined as typing aids.



Program Control Keys

The keys shown below allow you to control execution of the program stored in the computer's memory. Most of these keys are demonstrated in the remaining chapters of this manual.

RUN

RUN starts a program running from the beginning.

PAUSE

PAUSE pauses program execution after the current line.

CONTINUE

CONTINUE resumes program execution from the point where it was paused. It is also used like ENTER or EXECUTE to respond to a program prompt.

STOP
shift- CLR I/O

STOP stops program execution after the current line. Unlike PAUSE, you cannot resume execution of a program stopped with STOP by pressing CONTINUE. To restart the program, use the RUN key.

RESET
shift- PAUSE

RESET stops program execution immediately without erasing the program from memory. The `BASIC Reset` message indicates the computer is ready for your command.

CLR I/O

CLR I/O pauses program execution when the computer is performing or trying to perform an I/O operation. Press CLR I/O instead of PAUSE when the computer is hung up on an I/O operation, since PAUSE works only after the computer finishes the current program line. Pressing CLR I/O cancels the I/O operation and pauses the program at the current line.

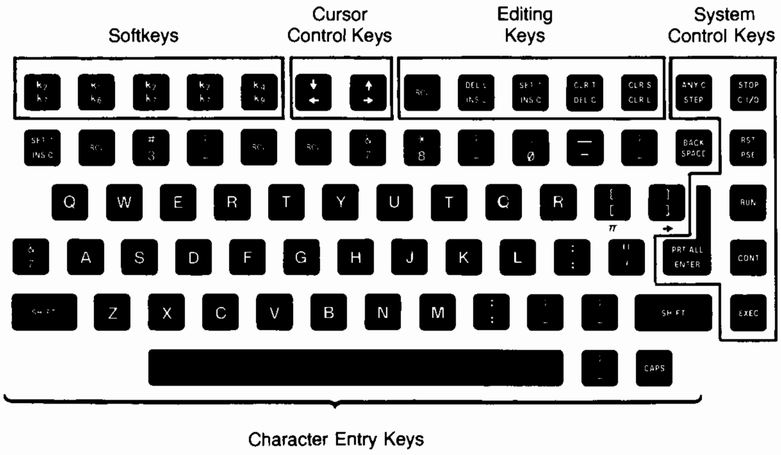
Now skip ahead to Chapter 5.

HP 98203A Keyboard Definitions

Note

If you have an HP 46020A or an HP 98203B keyboard, ignore this section and refer to one of the two preceding sections.

The keys on the HP 98203A keyboard are arranged into the following functional groups:



HP 98203A Keyboard

This section provides a handy reference guide to BASIC's key definitions for the HP 98203A keyboard. Keep in mind that other system programs may define the keys differently. Each key will be demonstrated where possible. One point to clarify: the **cursor** that we refer to in the following paragraphs is the blinking-underline that points to a location on the screen.

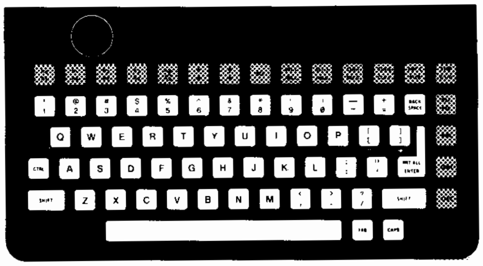
Note

Before you proceed, type:

```
SCRATCH (EXEC)
```

This clears the computer of any programs that might be left in memory from previous demonstrations.

Character Entry Keys



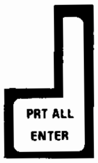
The character entry keys are arranged like a typewriter, but have some added features.

The (CAPS) key sets the unshifted keyboard to either upper case (which is the default after BASIC is booted) or lower case (normal typewriter operation). The computer displays which mode the computer is in when you press the (CAPS) key.

Type a few words, then press (CAPS) and continue typing. Notice the case change. Press (CLR L) when you're done.

You can enter standard upper-case and lower-case letters, using the (SHIFT) key to access the alternate case.

Type a few words, pressing (SHIFT) to change the case of the first letter of each word. Now press (CAPS) and continue typing. Notice that the alternate case that (SHIFT) accesses depends on the setting of (CAPS). Press (CLR L) when you're done.



The (ENTER) key has several functions:

1. When a running program prompts you for data, you respond by typing the requested data and then pressing (ENTER). This signals the program that you have provided the data and that it can resume execution. The (EXEC) key can also be used for this function.

2. When typing in lines of a program, the ENTER key is used to store each line of program code. The EXEC key can also be used for this function.
3. Like the EXEC key, the ENTER key can be used to execute commands and calculations.

Type `EDIT` and press ENTER. Notice the number `10` now displayed on the screen–this is the line number of the first line of a BASIC program. The computer is waiting for you to type in the line. Type:

```
!FIRST LINE
```

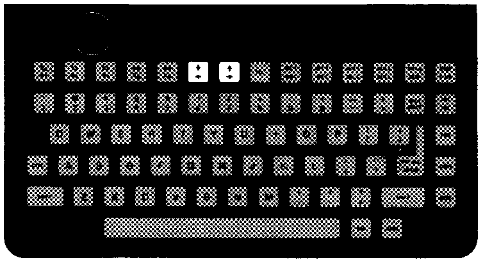
and press ENTER. Notice that the computer accepts the statement as a program line and displays `20` in preparation for the next one. Press PSE to exit.

The TAB key moves the cursor forward to preset tabs. Pressing SHIFT-TAB moves the cursor back to preset tabs.

Before TAB can be used, a tab must be set. Press the space bar to move the cursor forward a few spaces and press SET T. Move the cursor back several spaces using ←, then press 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 CLR T.

The CTRL (control) key works like SHIFT to access a set of standard control characters, such as line-feed and form-feed. These characters are useful to the programmer for controlling some devices and for communicating with other computers. You probably won't need them when running programs. The available control characters are listed in the Useful Tables section of the *BASIC 3.0 Language Reference.*

Cursor-Control Keys



The cursor-control keys move the display cursor. The ↑ and ↓ keys allow you to scroll lines in the output area up and down. The → and ← keys allow you to move horizontally along a line. The BACK SPACE key works just like the ← key.

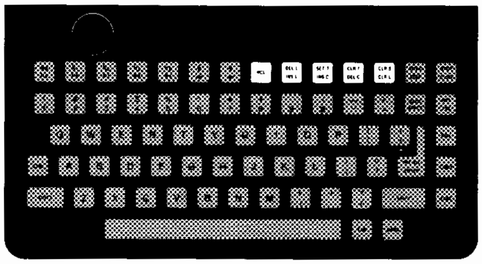
The cursor control wheel (also called the knob) allows you to rapidly scroll the output area up and down or move the cursor left and right, depending on the SHIFT key. With the SHIFT key depressed, the knob scrolls the output area up and down. Without the SHIFT key depressed, the knob moves the cursor left and right.

To test the horizontal movement of the cursor, type a few words and press the ← and → keys. Notice that the cursor cannot be moved beyond the characters you have typed. Now rotate the wheel to move the cursor. Press CLR L when you're done.

To test the vertical scrolling, type EDIT and press EXEC. 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
```

Now, press SHIFT and rotate the knob to scroll the text up and down. Also try out the ↑ and ↓ keys. When you're done, press PSE to exit.



Editing Keys

The editing keys put easy character editing and line editing at your fingertips. Some of these keys only work when you are in edit mode, which is entered by typing:

```
EDIT EXEC
```

Edit mode is described in detail in *BASIC 3.0 Programming Techniques*, Chapter 2. To exit edit mode, press PSE.

The RCL key recalls the last line that you entered, executed or deleted. Several previous lines can be recalled this way. RCL 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" EXEC
```

to print the number 1 on the screen. Now press RCL 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 EXEC again. Now press RCL several times to see all of the statements it remembers from the last entered to the earliest entered. Then press SHIFT-RCL several times to review the statements from the earliest to the last. Press CLR S to exit.

INS L inserts a new line above the cursor's current position (edit mode only).

Type EDIT EXEC. Type in this line (if it isn't already there):

```
10 !FIRST LINE
```

Now, with the cursor somewhere on line 10, press INS L. Notice that a new line number (1) is inserted before line 10. Press PSE to exit.

DEL L
INS L

DEL L deletes the line containing the cursor (edit mode only).

Type EDIT EXEC. Position the cursor to the line:

```
10 !FIRST LINE
```

and press DEL L. The line is removed. To restore it, press RCL to retrieve it, then ENTER to enter it into the program. Press PSE to exit.

SET T
INS C

INS C 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 INS C. Now type:

```
OF INSERT MODE
```

and press INS C 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 CLR L when you're done.

SET T
INS C

SET T sets a tab at the cursor's current position. Tabs are in effect for the keyboard line until cleared by either CLR T or the SCRATCH A statement. The SCRATCH commands are explained in *BASIC 3.0 Programming Techniques*, Chapter 2. To demonstrate SET T, see TAB.

CLR T clears a tab previously set at the cursor's position. To demonstrate CLR T, see TAB.

DEL C deletes the character at the cursor's position. Type a few words and experiment with DEL C, 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.

CLR L clears the keyboard line and message/results line. Type a few words and press CLR L to clear them.

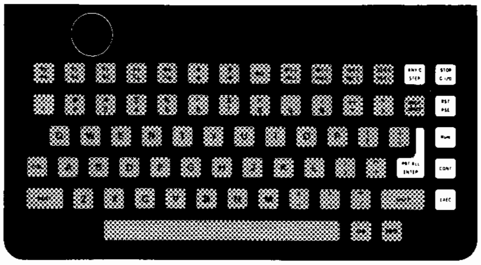
CLR S clears the entire alpha screen.

Type the following BASIC command:

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

Now press EXEC to execute it. Press RCL to recall the command and press EXEC again. Repeat this step several times to fill the screen with messages. Now press CLR S to erase all lines at once.

System Control Keys



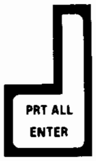
The keys on the right-hand side of the keyboard control various system functions related to the display, printer and editing operations. Most of these keys execute their functions immediately, as the key is pressed.

STEP allows the programmer to step through a program, one line at a time. Using the STEP key to debug programs is covered in *BASIC 3.0 Programming Techniques*, Chapter 12.

ANY C is used to find any ASCII character. First press ANY C Then enter a three-digit number from 000 thru 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 section of the *BASIC 3.0 Language Reference.*

Press ANY C, then type 189. 189 is the decimal equivalent of "§", which is now displayed in the keyboard line. Press CLR L to erase it.

RST (reset) stops program execution immediately without erasing the program or data from memory. The `BASIC Reset` message indicates that the computer is ready for your command.



The PRT ALL key turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device. Press PRT ALL once to set printall to ON and again to set printall to OFF. The printall mode is displayed in the lower-left corner of the screen.

The screen's output area is the default printall device at powerup. Selecting an external printall device is explained in *BASIC 3.0 Programming Techniques*, Chapter 8.

Press PRT ALL to turn on printall mode. Now type in the following command:

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

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 EXEC
```

Because this is not an executable BASIC statement, an error message is displayed, both at the bottom of the screen and in the printall area at the top. This way, a log is produced of all commands typed and executed at the keyboard, along with any error messages. Press CLR S to clear the screen, and press PRT ALL to turn off printall mode.

RUN starts a program running from the beginning.

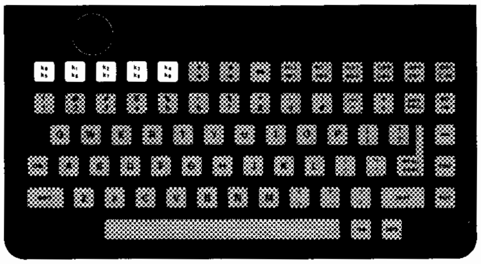
PSE pauses program execution after the current line. When in edit mode, PSE causes the computer to exit edit mode. Some BASIC keyboard commands cannot be executed while a program is running. In this situation, you can press PSE to suspend program execution, type and execute your keyboard command, then resume the program with the CONT key (described next). (Some keyboard commands make a program non-continuable.)

CONT resumes program execution from the point where it was paused. It may also be used like ENTER and EXEC to respond to a program prompt.

C I/O pauses program execution when the computer is doing an I/O operation. Press C I/O instead of PSE when the computer is hung up on an I/O operation, since PSE works only after the computer finishes the current statement. Pressing C I/O cancels the I/O operation and pauses the program at the current line.

STOP stops program execution after the current line. Unlike PSE, you cannot resume execution of a program stopped with STOP by pressing CONT. To restart the program from the beginning, use the RUN key.

Softkeys



The ten keys labeled (k0) thru (k4) (using the (SHIFT) key) and (k5) thru (k9) are defined under program control. The program may also display a label for each defined key. Pressing a defined key tells the computer to interrupt whatever it's doing and start running the designated part of the program.

We call these keys "softkeys" because the program or "software" defines and labels them.

With the KBD BIN file loaded, softkeys can be used as typing aids.

Working With BASIC

Chapter 5

You now have BASIC loaded and have become familiar with your keyboard. In this chapter you will continue to learn how BASIC works by writing, running, editing, storing, and loading a simple program. You will also learn the BASIC arithmetic operations. At the end of this chapter are some hints on dealing with errors and interpreting error messages.

Programming With BASIC

> **Note**
>
> When entering program lines and doing other operations, you must press certain keys. The keys differ depending on your keyboard. Keys for all three keyboards are shown in this order: HP 46020A, HP 98203B, and HP 98203A
> (e.g., Return, EXECUTE, or EXEC).
>
> Press the key that is on **your** keyboard.

Writing A Program

Type:

`SCRATCH` Return, EXECUTE, or EXEC

This erases any program that may have been in your computer's memory. You now have a "blank slate".

Type:

`EDIT` Return, EXECUTE, or EXEC

The number `10` should appear in the middle left of your display, indicating the computer is ready to accept the first program line.

Type:

`For A=0 to 99` Return, ENTER, or ENTER

The computer accepts line 10 and presents line `20` so you can type in the second statement.

Continue the process until you have the following program on your screen:

```
10    FOR A=0 TO 99
20    PRINT A
30    NEXT A
40    END
50
```

You have now written a program that will print all numbers from 0 to 99, then stop.

Running Your Program

Note

If you have an HP 46020A keyboard, you must be in System mode with the System softkeys displayed in order to use the **RUN** softkey (*f3*). Otherwise, you can type RUN Return.

Press: **RUN** (*f3*), RUN, or RUN

Numbers up to 99 are printed at the left of the display.

Now let's pause the program before it stops.

Press: **RUN**, RUN, or RUN
and then immediately press unshifted-Stop, PAUSE, or PSE

The program pauses after executing the current line, displays the next line to be executed, and displays the paused indicator (-) at the lower-right of the screen.

You can now step through the program one line at a time by pressing **Step** (*f1*) STEP, or STEP. You can also continue running the program by pressing **Continue** (*f2*), CONTINUE, or CONT. If you press **RUN**, RUN, or RUN while paused, the program begins running from the beginning.

Editing Your Program

Now suppose you want to change your program so it counts from 0 to 199.

Type:

`EDIT` Return, EXECUTE, or EXEC

Your program should appear on the screen.

Use your ▼, ↓, or ↓ key, your cursor control wheel, or the mouse to place the cursor at line `10`. (You could accomplish the same result by typing `EDIT 10` Return, EXECUTE, or EXEC.)

Edit line `10` by changing `99` to `199`. Remember to press

Return, ENTER, or ENTER

to successfully change the program statement.

Press: **RUN**, RUN, or RUN to see if your new program works.

Storing And Loading Your Program

Verify that your program is now in your computer's memory.

Type:

`LIST` Return, EXECUTE, or EXEC

If someone turns your computer off now, your program will be lost. The computer's memory goes blank when it is turned off. Therefore, you want to store your program where it will be safe and can be used again and again.

Store your program on the same drive from which you booted your system. Insert a blank, initialized disc into the drive.

Type:

`STORE "COUNT"` Return, EXECUTE, or EXEC

Your program, named COUNT, is now stored on your system disc.

Now, erase your program from computer memory.

Type:

`SCRATCH` (Return), (EXECUTE), or (EXEC)

Verify that your program is no longer in computer memory.

Type:

`LIST` (Return), (EXECUTE), or (EXEC)

The display should just show available memory. The program should be gone.

Let's **load** the program from the system disc.

Type:

`LOAD "COUNT"` (Return), (EXECUTE), or (EXEC)

Press: **RUN**, (RUN), or (RUN)

The program should have been loaded into your computer and should run normally.

Arithmetic Operations

The BASIC language system lets you perform arithmetic calculations like a typical hand-held calculator. Just type in an arithmetic expression, press Return, EXECUTE, or EXEC, and your computer will deliver the answer.

Some of the arithmetic operators used by your computer may differ from those you're familiar with. Look over the following table of operators, then we'll try out some examples:

Operator	Symbol	Key(s)
Exponentiation	^	Shift-–, ^, Shift-6
Division	/	/
Multiplication	*	*, *, or Shift-8
Subtraction	-	–
Addition	+	+, +, or Shift-=

Arithmetic Examples

Before you perform arithmetic operations, clear the display, if necessary, by pressing Clear display, CLR SCR, or CLR S. Also, never insert spaces between the digits of a number.

Try It Yourself

Problem 1: If a computer's screen can display 25 lines, and each line can be up to 80 characters long, how many characters can be displayed on the computer's screen?

Type:

```
80*25
```

Return, EXECUTE, or EXEC

Your computer returns the answer: `2000` characters.

Problem 2: If you spend 3% of your time today reading this manual, how much of your eight-hour workday is left for work?

Type:

`8-8*.03` Return, EXECUTE, or EXEC

Your computer returns the answer: `7.76` hours for work.

Problem 3: If the floor in your office is square, with each side measuring 6.2 metres, how many square metres of carpeting are needed to cover it? You can either multiply 6.2 * 6.2, or find the square of 6.2 by raising it to the second power (6.2^2).

Type:

`6.2^2` Return, EXECUTE, or EXEC

Your computer returns the answer: `38.44` square metres.

You can always use **RECALL,** RECALL, or RCL to recall an expression for additional computations. Press the recall key to recall the last problem and add 2 to the expression by typing:

`+2` Return, EXECUTE, or EXEC

Your computer returns the answer: `40.44`.

Your computer saves time by automatically assuming INTEGER notation whenever you enter small numbers without a decimal point, or an "E" to indicate an exponent. But INTEGER notation has its limits: from −32 768 to +32 767. So if you execute an expression with results beyond this integer range, say:

`126*261` Return, EXECUTE, or EXEC

The computer reports:

```
ERROR 20  INTEGER overflow
```

To avoid this error, be sure to enter a decimal point in at least one of the numbers. This causes the computer to use REAL notation.

Try It Yourself

Type:

126.*261 (Return), (EXECUTE), or (EXEC)

Your computer reports the answer, 32886.

Now try multiplying some really large numbers.

Type:

6000000*90000 (Return), (EXECUTE), or (EXEC)

The computer reports the answer, 5.4E+11, in scientific notation. When the absolute value of the result cannot be represented in six digits (result >= 1E6), the computer displays it in scientific notation. The number after the "E" is the exponent. It indicates how many places the decimal point must be moved to the right of the mantissa. In this case, the number is 540 000 000 000.

Arithmetic Hierarchy

When an arithmetic operation having more than one operator is executed, the computer evaluates the expression according to a predefined hierarchy:

^	(exponentiation)	performed first
*/	(multiplication and division)	
+-	(addition and subtraction)	performed last

As an expression is scanned, the highest-priority operations are performed first. The results of these operations are used as inputs to the lower-priority operators. Equal-priority operations are performed from left to right. When you want to be certain that one operation in the expression is evaluated before another, place that operation in parentheses.

Try It Yourself

Type in Problem 2 from the previous demonstration again:

`8-8*.03` (Return), (EXECUTE), or (EXEC)

The computer responds with the answer: `7.76`. The multiplication (8*.03) was evaluated first. By enclosing the subtraction in parentheses, you can change the order of evaluation and thus the result.

Type:

`(8-8)*.03` (Return), (EXECUTE), or (EXEC)

The computer now returns the answer: `0`.

The following example shows the order of evaluation for an expression with various operators:

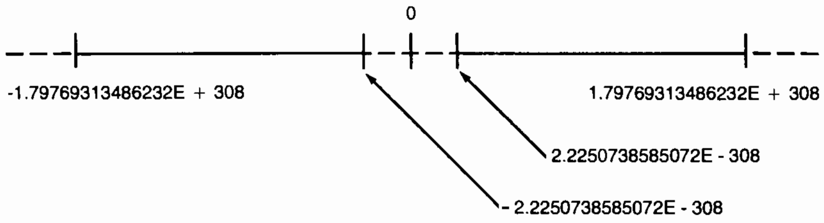
	2+3*6/(7-4)^2
1. evaluate parentheses:	2+3*6/**(7-4)**^2
2. exponentiation:	2+3*6/**3^2**
3. multiplication:	2+**3*6**/9
4. division:	2+**18/9**
5. addition:	**2+2**
result:	**4**

If you nest parentheses, the inner-most operation is evaluated first.

Whenever you're in doubt about the order of execution, use parentheses to force the order you want. However, using parentheses for "implied" multiplication is not allowed. So the expression, 4(5−2) must be entered as `4*(5-2)`.

Computing Range

The range of values that your computer can handle is from about 10^{-308} thru 10^{308}. More exact limits are:



Computing Range

Significant Digits and Rounding

Your computer can input, store, and calculate numbers having approximately 15 significant digits. Significant digits are those which determine a number's actual value: leading zeros are not counted. When you enter a number with more than 16 significant digits, the excess digits are truncated (dropped) and the 16th digit is rounded before the number is accepted. The computer automatically rounds each final result to 12 digits before it's output. This can be decreased or increased up to 15 digits by specifying an image in your output operations. For details on printing with images, refer to your *BASIC 3.0 Programming Techniques* manual, Chapter 8.

The computer rounds a number by first looking at the 13th digit. If it's 5 or greater, the 12th digit is incremented by one. If the 13th digit is less than 5, it's simply truncated from the number. This is called "rounding up."

BASIC functions are available for rounding numbers before processing or storage. For details on these functions, consult your *BASIC 3.0 Programming Techniques*, Chapter 4.

Dealing with Errors

As you saw in some of the earlier demonstrations, the BASIC system responds to an incorrect operation with an error message. The message contains an error number and a short description of the error. (If the ERR BIN file is not loaded, only the error number is displayed.) If the error occurred in a running program, the number of the offending line is also given. These error messages may seem short and cryptic, but it's important for you to understand what the computer is trying to say. If you're unfamiliar with error messages, take a few minutes to read this section.

Keyboard Error Messages

The computer rejects incorrect keyboard entries by beeping and displaying an error message.

Try It Yourself

Suppose you want to calculate 10/.03, but instead you typed:

`10/.0` Return, EXECUTE, or EXEC

The computer knows division by zero is illegal and politely reminds you of this fact with the message:

```
ERROR 31  Division by 0 or X MOD 0
```

As another example, what would happen if you meant to type the BASIC command CAT, but typed the following line by mistake:

`CAR` Return, EXECUTE, or EXEC

The computer knows this is not a legal BASIC command, and displays the following error message:

```
ERROR 910  Ident not found in context
```

Whenever you make this type of error, the computer positions the cursor under the word it doesn't recognize.

To understand the meaning of error messages, you need to look at the error as the computer sees it–not as you *expected* to see it. This is not always easy to do. Fortunately, unless you're writing programs, you don't need to understand every error message.

Most keyboard error messages simply point out typing mistakes, such as CAR instead of CAT, or a missing quote mark. When an error occurs, carefully check your spelling and punctuation, make any corrections, and execute the line again. Remember, computers are very narrow-minded owing to their limited intellects, so you must humor them with extremely precise instructions. Don't let error messages bother you; they're a normal part of the computing experience.

Program Errors

All the rules for keyboard commands–and many more–apply to running a program. When an error occurs during execution of a program, the computer pauses and displays an error message like this:

```
   ERROR 31 IN 280  Division by 0 or X MOD 0
```

This error code (31) refers to an attempted division by 0 in program line 280. There are many different error messages, with expanded definitions, listed in the back of both the *BASIC 3.0 Programming Techniques* and the *BASIC 3.0 Language Reference* manuals. Some represent programming mistakes; others represent mistakes you have made while entering data. The latter case probably results from a simple typing mistake on the input line, and the computer is waiting for you to re-enter the correct data. No matter what the error message, you should get the programmer involved.

There is one golden rule to remember whenever you get an error message while running a program: **Any time you get a computer error message from a running program, it is the programmer's fault–not yours!** This is always true unless, of course, the programmer is you.

The programming language on your computer allows a programmer to anticipate or intercept any error resulting from an operator's mistake. Admittedly, such a comprehensive level of error-catching is difficult and time-consuming to achieve. The programmer may not have taken the time to deal with your particular error, but the interception of all errors is an admirable goal. The program should display an explanation of the error, suggest corrective action, and give you a second chance. If it doesn't, write down the error message and contact the person who is supporting your program. Tell them what you did, show them the error message, and act like you expect them to do something about it.

Talking To Peripherals With BASIC

Chapter

6

This chapter introduces some fundamental disc drive and printer operations that you should know about. We'll show you how to get your discs ready to store data and programs, how to copy the contents of one disc to another, and how to list a disc's directory of all files that it contains. We'll also demonstrate how to generate a printed program listing and how to "dump" graphics and alpha displays from the screen to the printer.

Simple Disc Drive Operations

In this section, some simple BASIC disc drive operations are described and demonstrated. But before you can access your disc drive and try some of these techniques, you must determine your drive's **mass storage unit specifier**. Don't worry, this sounds a lot worse than it is.

In plain English, the mass storage unit specifier (or **msus**, for short) is what your computer uses to identify your disc drive. To identify the drive to the computer, you need to specify three things: what is it, where is it, and what part of it you want to access. Let's take each of these separately.

What Is It?: The Device Type

The component of the msus that tells the computer what type of device it's dealing with is called the **device type**. When you provide the device type, the computer knows the storage capacity of the device (how many bytes it can hold), its directory structure (how it finds information on the disc, like using a telephone directory to find someone's phone number), and other information it needs in order to access the device.

The device type is often just the model number of the drive. For instance, the device type of the HP 82901 is just `HP82901`. Sometimes, the device type is the model number of *another* disc drive. For example, the device type of the HP 9121 is `HP8290X`. This is because, from the computer's point of view, the 9121 looks like an HP 82901 or 82902 disc drive. At other times, the device type is a description of the class of devices the drive belongs to. For example, the HP 7908 is a Command Set 80 (CS/80) device, and its device type is, therefore, `CS80`.

Note

The device type is not required for most devices when specifying the msus. The device selector and unit number are normally enough to specify the msus. The examples in this chapter clarify this.

Where Is It?: The Device Selector

The component of the msus that tells the computer where your device is located is called the **device selector**. The device selector always contains the select code of the interface your drive is connected to, and if you have an HP-IB device, it also contains the primary address of the device. With this information, the computer knows exactly where the device is located.

HP-IB Device Selectors. The device selector of a drive connected to an HP-IB interface is built from the select code of the HP-IB and the primary address of the drive. To derive the device selector, multiply the interface select code by 100 and add the drive's primary address. For example, if the drive is connected to the internal HP-IB (select code 7) at primary address 0, the device selector would be (7*100) + 0 = 700.

Non-HP-IB Device Selectors. If the disc drive you want to access is connected to an interface other than the HP-IB, the device selector is simply the select code of the drive's interface. For example, a disc drive connected to a GPIO interface at select code 12 will have `12` as its device selector.

Which Part of It?: The Unit Number

Mass storage devices are partitioned into **units**. Most of the time, a unit is the same as a drive. For example, the HP 82901 and HP 9121 have two disc drives: the one on the left is unit 0 and the one on the right is unit 1. In other cases, unit numbers identify different portions of one disc, as in the HP 9133A's hard (Winchester) disc, which is assigned unit numbers 0 thru 3.

Finding Your MSUS

Now it's time to put the components together to form the msus for your device. The syntax of the msus is illustrated below:

:device type,device selector,unit number

The unit number is optional; if left unspecified, it is assumed to be 0. The msus must be enclosed within **double quotes** when used in a BASIC command.

The easy way to find the msus of your *flexible* disc drive is to let your computer tell you what it is. Remember, when the computer boots a system, it displays the location of the disc drive where it found the system in the upper-right corner of the screen. This is the msus of that device. Chapter 2, under the heading "Booting Basic" provides the procedure for finding the msus for each device in your system.

Another way to find your msus is to use the following table. It gives the device type, factory-set primary address, and unit number(s) of each Series 200 supported disc drive.

Find your drive in the table. If you haven't changed the factory settings for primary address and unit number, you can use the device type, primary address, and unit number exactly as shown.

HP	msus	Factory Settings:	
Device	Device Type	Address	Unit No.
HP 7908 (h)	CS80	0	0
HP 7908 (t)	CS80	0	1
HP 7911 (h)	CS80	0	0
HP 7911 (t)	CS80	0	1
HP 7912 (h)	CS80	0	0
HP 7912 (t)	CS80	0	1
HP 7914 (h)	CS80	0	0
HP 7914 (t)	CS80	0	1
HP 7933H	CS80	0	0
HP 7935H	CS80	0	0
HP 9121S	HP8290X or HP9121	0	0
HP 9121D (l)	HP8290X or HP9121	0	0
HP 9121D (r)	HP8290X or HP9121	0	1
HP 9122 (l)	CS80	0	0
HP 9122 (r)	CS80	0	1
HP 9130K (f)	INTERNAL	4	0 or 1
HP 9133 (f)	HP8290X or HP913X	1	0
HP 9133A (h)	HP9895 or HP913X	0	0 thru 3
(Opt. 010 h)	HP9133 or HP913X	0	0
HP 9133B (h)	HP9133 or HP913X	0	0
HP 9133V (h)	HP9133 or HP913X	0	0
(Opt. 004 h)	HP9895 or HP913X	0	0 thru 3
HP 9133XV (h)	HP9133 or HP913X	0	0
(Opt. 010 h)	HP9133 or HP913X	0	0
HP 9134A (h)	HP9895 or HP913X	0	0 thru 3
(Opt. 010 h)	HP9134 or HP913X	0	0
HP 9134B (h)	HP9134 or HP913X	0	0
HP 9134XV (h)	HP9134 or`HP913X	0	0
(Opt. 010 h)	HP9134 or HP913X	0	0
HP 9135 (f)	HP8290X or HP913X	1	0
HP 9135 (h)	HP9895 or HP913X	0	0 thru 3
(Opt. 010 h)	HP9135 or HP913X	0	0
HP 9895M (l)	HP9895	0	0
HP 9895M (r)	HP9895	0	1
HP 9895S (l)	HP9895	0	2
HP 9895S (r)	HP9895	0	3
HP 82901M (l)	HP82901	0	0
HP 82901M (r)	HP82901	0	1
HP 82901S (l)	HP82901	0	2
HP 82901S (r)	HP82901	0	3
HP 82902M	HP82902	0	0
HP 82902S	HP82902	0	2
HP 9885M/S	HP9885	NA	0 thru 3

(h) indicates a hard disc drive
(f) indicates a flexible disc drive
(t) indicates a tape drive
(l) indicates the left drive of a dual drive machine
(r) indicates the right drive of a dual drive machine

Examples

The following msus describes the left drive of an 82901S connected to the internal HP-IB (select code 7). The 82901S is set to primary address 0, and the left drive has a unit number of 2.

```
":HP82901,700,2" or ":,700,2"
```

The next msus designates the left drive of a 9121 connected to the internal HP-IB (select code 7). The primary address of the 9121 has been set to 7, and the unit number of the left drive is 0.

```
":HP8290X,707" or ":HP9121,707" or ":,707"
```

Note that the unit number need not be specified because it defaults to 0.

Note

If you want additional information on the msus and other mass storage topics, refer to your *BASIC 3.0 Programming Techniques*, Chapter 7.

Initializing Discs

Refer to Chapter 3 for disc initialization procedures.

Copying Discs

Refer to Chapter 3 for disc copying procedures.

Listing a Disc's Directory

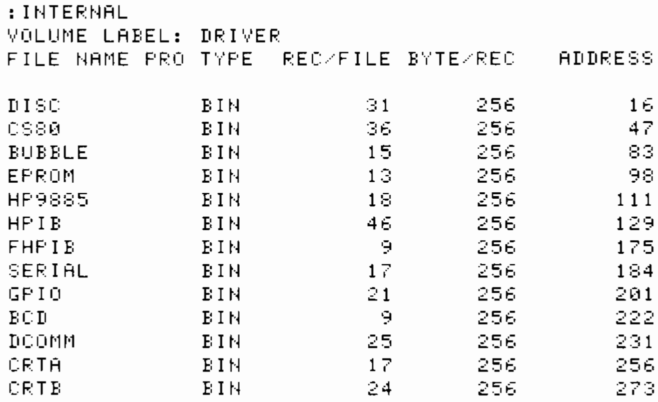
You can find out a lot of information about the files you have on a disc by listing the disc's directory with the CAT command.

Try It Yourself

Insert the *Drivers Disc* supplied with your BASIC language system into one of your disc drives. Execute the CAT command with the msus of the drive containing the disc. For example, if the disc is inserted into the left drive (unit 0) of an HP 82901M connected to the internal HP-IB (select code 7) at primary address 0, the CAT command would look like this:

```
CAT ":HP8290X,700,0"
```

The resulting display should look something like this:



```
:INTERNAL
VOLUME LABEL: DRIVER
FILE NAME PRO TYPE  REC/FILE BYTE/REC   ADDRESS

DISC          BIN         31      256        16
CS80          BIN         36      256        47
BUBBLE        BIN         15      256        83
EPROM         BIN         13      256        98
HP9885        BIN         18      256       111
HPIB          BIN         46      256       129
FHPIB         BIN          9      256       175
SERIAL        BIN         17      256       184
GPIO          BIN         21      256       201
BCD           BIN          9      256       222
DCOMM         BIN         25      256       231
CRTA          BIN         17      256       256
CRTB          BIN         24      256       273
```

Example of a Disc Directory

All of the files on your *Drivers Disc* are file type BIN. BIN files hold system programs coded in a binary (machine language) format. BIN files are loaded into memory with the LOAD BIN command.

Other file types are stored on discs:

- PROG files hold programs that are stored on disc with the STORE command and loaded into memory with the LOAD command.
- ASCII files contain either programs or other information coded in a universal data format so that they can be read by other computers. ASCII files are stored on disc with the SAVE command and are brought into memory with the GET command.

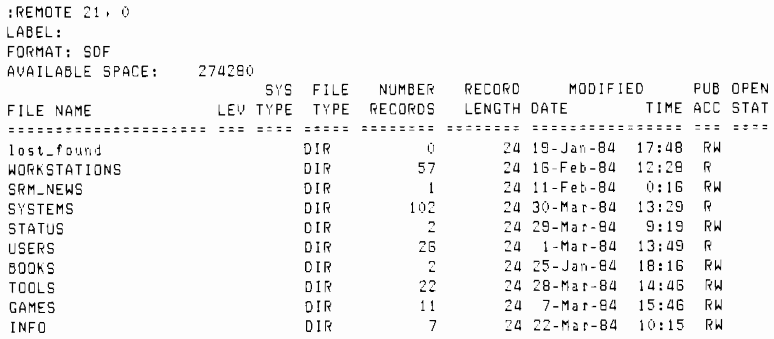
- SYSTM files contain system programs that can be booted when the power is turned on.
- BDAT files contain data used by a BASIC program.

If the file type is a number, rather than a word, it indicates that the file type is not compatible with the BASIC language system. These files can be cataloged, but the contents of the file cannot be accessed by the computer.

The directory tells you more than just each file's name and type: it also indicates each file's size and location on the disc. File sizes are of great use to a programmer who must organize and support each file.

SRM Disc Files

If your files are located on a Shared Resource Management disc, the CAT display will look slightly different as shown below:



```
:REMOTE 21, 0
LABEL:
FORMAT: SDF
AVAILABLE SPACE:    274280
                          SYS  FILE    NUMBER   RECORD      MODIFIED    PUB OPEN
FILE NAME             LEV TYPE TYPE   RECORDS   LENGTH DATE        TIME ACC STAT
===================== === ==== ===== ======== ======== ================ === ====
lost_found                     DIR          0       24 19-Jan-84  17:48 RW
WORKSTATIONS                   DIR         57       24 16-Feb-84  12:28 R
SRM_NEWS                       DIR          1       24 11-Feb-84   0:16 RW
SYSTEMS                        DIR        102       24 30-Mar-84  13:29 R
STATUS                         DIR          2       24 29-Mar-84   9:19 RW
USERS                          DIR         26       24  1-Mar-84  13:49 R
BOOKS                          DIR          2       24 25-Jan-84  18:16 RW
TOOLS                          DIR         22       24 28-Mar-84  14:46 RW
GAMES                          DIR         11       24  7-Mar-84  15:46 RW
INFO                           DIR          7       24 22-Mar-84  10:15 RW
```

Example of an SRM Directory

Note that the SRM CAT display shows Public Access instead of Protection status and has an additional column for open status. These features are described thoroughly in the *SRM HP Series 200 Workstation Manual.*

Copying Files

You created backup discs by copying the entire contents of one disc to another disc. You can also copy a specific file on one disc to another disc. The COPY statement is used to copy files. For example, suppose you wanted to copy a file named COUNT from the disc in the right-hand drive of a Model 236 to a file named SUMS on the disc in the left-hand drive. You would type:

```
COPY "COUNT:INTERNAL,4,0" TO "SUMS:INTERNAL,4,1"
```

The file named COUNT is written from one disc to the other and assigned the new name SUMS. The destination file **can** be assigned the same name as the source file.

Notice that when copying files, the COPY statement requires a filename before the colon (`:`) in both the source and destination sides of the statement. When copying entire discs, the filenames are deleted.

The MASS STORAGE IS Statement

The MASS STORAGE IS statement (or MSI, for short) allows you to designate a disc drive as the system default mass storage device. Once this is done, all mass storage operations automatically refer to the default drive, and you don't have to type the msus each time.

> **Note**
>
> If you booted BASIC from a disc drive, the drive it was booted from automatically becomes the default mass storage device. The MASS STORAGE IS statement is only needed when you want to *change* the current default mass storage device.

To designate a drive as the system default, type:

`MSI "`**msus**`"`

Substitute your drive's msus in the line above, and press (Return), (EXECUTE), or (EXEC).

For example, to designate unit 1 (the right-hand drive) of an HP 9121 disc drive connected to the internal HP-IB at primary address 0 as the system default, type:

`MSI ":HP8290X,700,1"` or `":,700,1"` (Return), (EXECUTE), or (EXEC)

Now you can just type and execute `CAT` to list this drive's directory, for example. No further reference to the msus is required for most BASIC statements (INITIALIZE is an exception–you must use the msus). If, however, you want to reference a disc drive other than the system default, you must use its msus.

Try It Yourself

Execute an MSI statement as shown previously for one of your disc drives. Now insert any BASIC disc into this drive and type:

CAT (Return), (EXECUTE), or (EXEC)

The default drive is automatically accessed and the directory is listed.

Simple Printer Operations

In this section, some simple BASIC printer operations will be described and demonstrated. Like disc drives, printers must also be declared to the system before they can be used. Fortunately, the method used to define the location of a printer is much simpler than the msus used for disc drives.

Finding the Device Selector

When accessing a printer, you need only specify its device selector. This is identical to the device selector portion of the msus used to access disc drives.

HP-IB Printers. The device selector of a printer connected to the HP-IB interface is formed from the select code of the HP-IB and the primary address of the printer. To derive the device selector, multiply the interface select code by 100 and then add the printer's primary address. For example, if the printer is connected to the internal HP-IB (select code 7) at primary address 1, the device selector would be (7*100) + 1 = 701.

All HP-IB printers are set to primary address 1 at the factory. If you have not changed this setting, the device selector for your printer is 701.

Non-HP-IB Printers. The device selector of a printer connected to an interface other than the HP-IB is simply the select code of its interface.

The PRINTER IS Statement

The PRINTER IS statement is similar to the MASS STORAGE IS statement. It designates a printer as the system default printer, and all printer operations are directed to it.

The PRINTER IS statement is very simple to use. Just type `PRINTER IS` followed by the device selector of your printer (or `PRT` if you have just one printer). For example, to assign a printer on the internal HP-IB (select code 7) at primary address 1 as the default system printer, type:

`PRINTER IS 701` Return, EXECUTE, or EXEC

or

`PRINTER IS PRT` Return, EXECUTE, or EXEC

if you have just one printer.

> **Note**
>
> The computer's screen is also considered a "printer". When you type `PRINTER IS 1` or `PRINTER IS CRT` Return, EXECUTE, or EXEC you set the system default printer to the screen, and all output is directed to it. The screen is automatically assigned as the system printer when BASIC is booted.

The PRINT Statement

The PRINT statement is the primary means of sending characters to the system default printer. The PRINT statement syntax is simply `PRINT` followed by a list of things to be printed, separated by commas or semicolons. The options available with the PRINT statement are discussed in detail in *BASIC 3.0 Programming Techniques*, Chapter 8. In this book, we limit our discussion to sending simple messages to the printer.

Try It Yourself

Designate your printer as the system default by executing the appropriate PRINTER IS statement. Now, to send a message to the printer, just enclose the message in double quotes and type it following PRINT. For example, type:

```
PRINT "THIS IS A TEST OF THE SYSTEM DEFAULT PRINTER"
```

Return, EXECUTE, or EXEC

The message is printed on the printer. (If nothing was printed, make sure you have declared the printer as the default with the PRINTER IS statement. Also check that the printer is turned on and "on line".)

Type any other messages you want to print. Then restore the computer's screen as the default printer by typing:

`PRINTER IS CRT` Return, EXECUTE, or EXEC

The Printall Printer

As discussed in the last chapter, a record of all keyboard commands and error messages can be logged to a printall printer. When you press **Print All,** PRT ALL, or PRT ALL, you activate printall mode.

When the computer is turned on, the printall printer is automatically set to the computer's screen. To make an external printer the printall printer, use the PRINTALL IS command.

The PRINTALL IS command works just like the PRINTER IS statement. Simply type `PRINTALL IS` followed by the device selector of your printer and press Return, EXECUTE, or EXEC. For example, to set a printer on the internal HP-IB (select code 7) at primary address 1 to be the printall printer, just type:

`PRINTALL IS 701` Return, EXECUTE, or EXEC

Now when printall mode is activated with the **Print All,** PRT ALL, or PRT ALL key, all commands are logged on the printer. Try doing a CAT on your disc drive, and a PRINT on the default printer. Notice that each command you execute is printed.

Getting a Program Listing

To get a listing of a program, you must first LOAD it into your computer's memory. Then you can use the LIST command to print it.

If you simply type:

`LIST` (Return), (EXECUTE), or (EXEC)

the listing will be printed on the default system printer. If you want the listing to go to a printer other than the default, type LIST followed by # and the device selector of the printer. For example, to send the listing to a printer on the internal HP-IB (select code 7) at primary address 1, you would type:

`LIST #701` (Return), (EXECUTE), or (EXEC)

Try It Yourself

To get a listing of the Breakout game, follow these steps:

1. Assign one of your disc drives as the default mass storage device with a MASS STORAGE IS statement. Insert the *Manual Examples Disc* into that drive and close the door.
2. Type:

   `LOAD "BREAKOUT"` (Return), (EXECUTE), or (EXEC)

   to load the program into your computer. Wait for the run light to go out.
3. Set your printer to default system printer with the PRINTER IS statement.
4. Type:

   `LIST` (Return), (EXECUTE) or (EXEC)

   to list the program on your printer.

Dumping to a Printer

Note

You cannot dump graphics to an HP 82905 printer with the techniques described in this section. However, there is an HP 82905B Dump Graphics Subprogram (82905BDUMP) on your *Utilities Disc 2*. See the *BASIC 3.0 Utilities Library* manual for details.

You can "dump" alpha and graphics displays from your computer's screen to a printer by executing BASIC commands. To dump the alpha display, type:

`DUMP ALPHA` (Return), (EXECUTE), or (EXEC)

Note

You must have the GRAPH BIN file loaded in order to execute `DUMP GRAPHICS` and `DUMP DEVICE IS`, and to run the demonstration.

To dump the graphics display, type:

`DUMP GRAPHICS` (Return), (EXECUTE), or (EXEC)

Note

If you have a Model 237 computer, this statement dumps the combined alpha and graphics display.

Before you can dump alpha and graphics to your printer, you may need to designate your printer as the dump device with the DUMP DEVICE IS statement. If your printer is connected to the internal HP-IB at primary address 1, it is already set as the system default dump device when your computer is turned on.

If your printer's device selector is not 701, you must execute the DUMP DEVICE IS statement. Type `DUMP DEVICE IS`, followed by the device selector of your printer, then press (Return), (EXECUTE), or (EXEC). For example, if your printer is connected to the internal HP-IB (select code 7) at primary address 2, you would type:

`DUMP DEVICE IS 702` (Return), (EXECUTE), or (EXEC)

Try It Yourself

This demonstration assumes you have a graphics printer.

1. Set one of your disc drives to the system default with the MASS STORAGE IS statement. If not already done, boot BASIC using the instructions is Chapter 2. Insert the *Manual Examples Disc* into the system default drive.
2. If necessary, designate your printer as the dump device with the DUMP DEVICE IS statement.
3. Type:

   `LOAD "SRM",1` (Return), (EXECUTE) or (EXEC)

   to load a program that demonstrates the HP Shared Resource Management network.
4. The initial display of the SRM program is an alpha display.

   Type:

   `DUMP ALPHA` (Return), (EXECUTE), or (EXEC)

   to print a copy of this display on your printer. Press the **CONTINUE** softkey ((f5) or (k0)) to move on to the next display.
5. The next display is a graphics display. Type:

   `DUMP GRAPHICS` (Return), (EXECUTE), or (EXEC)

   to print a copy of the display on your printer.

Special BASIC Features

Chapter 7

This chapter explores some of the more impressive features of BASIC on your computer. We'll demonstrate several ways to use graphics, and show you how to dress up your alphanumeric displays with display-enhancement characters. You'll also learn how to make your computer do two things at once, and how you can interact with a running program. Even if you have no intention of programming in BASIC, you'll still enjoy trying the demonstrations and seeing some of the "tricks" your computer can do.

Note

As in previous chapters, key presses for all keyboards are provided. The order in which the keys are listed is: HP 46020A, HP 98203B, HP 98203A. Press the key listed for your keyboard.

The following demonstrations assume you have booted BASIC according to the instructions given in Chapter 2, and designated one of your disc drives as the default mass storage device with the MASS STORAGE IS statement (refer to Chapter 6). The display should also be the default system printer; if you're not sure that it is, type:

`PRINTER IS CRT` **Return**, **EXECUTE**, or **EXEC**

One other point to watch out for: many of these demonstrations ask you to transfer programs from disc into memory using the LOAD statement. If you execute a LOAD statement and get an "ERROR 120: Not allowed while prog running", press unshifted-**Stop**, **PAUSE**, or **PSE** or **Shift**-**Stop**, **STOP**, **STOP** and try it again. (Use the **Recall**, **RECALL**, or **RCL** key to recall the statement, rather than typing it over.)

Graphics and Alpha

Note

The Model 237 computer has a combined alpha and graphics display that cannot be separated. Therefore, you cannot do the following demonstration. Continue with "Display-Enhancement Characters".

BASIC, in effect, defines your computer to have *two* display screens: one for alphanumeric characters, the other for graphics. Each display can be presented on the screen individually, or they may both appear together.

If you have the HP 98203A keyboard, you can turn the displays on and off by executing BASIC commands. When you execute the GRAPHICS ON statement, the graphics display is turned on; executing GRAPHICS OFF turns it off. When you execute the ALPHA OFF statement, you turn off the alphanumeric display; pressing any key turns it back on again.

If you have the HP 98203B or HP 46020A keyboard, you have keys to turn the two displays on and off. The HP 98203B keys are labeled on the keyboard; the HP 46020A keys are unlabeled system control keys (refer to Chapter 4). The (ALPHA) key turns on the *alphanumeric* display if you press it once; turns *off* the *graphics* display if you press it a second time. The (GRAPHICS) key turns *on* the *graphics* display if you press it once; turns *off* the *alphanumeric* display if you press it again.

Try It Yourself

Note

The GRAPH BIN file must be loaded in order to do the following demonstration.

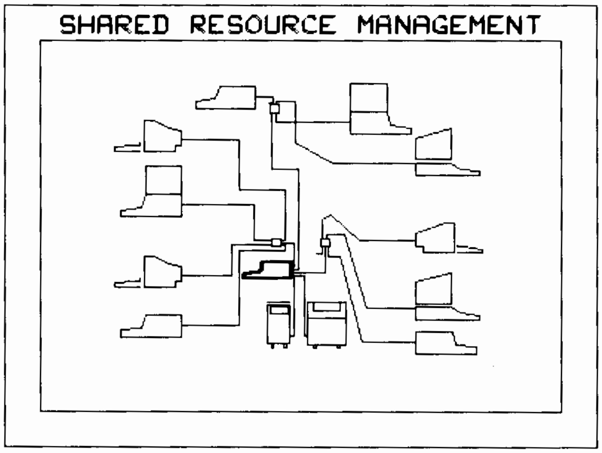
Insert the *Manual Examples Disc* into the default drive and close the door.

Now type:

```
LOAD "SRM",1
```

Return, EXECUTE, or EXEC

to load a program that demonstrates the HP Shared Resource Management network. Press **CONTINUE** as directed and wait for the following diagram to appear on the screen.



SRM Display

Press unshifted-Stop, PAUSE, or PSE to suspend execution of the program. Press the ALPHA key or type:

```
GRAPHICS OFF
```

Return, EXECUTE, or EXEC

and watch the graphics display disappear.

Now that the graphics display has something in it (even though you can't see it at the moment), let's put something in the alphanumeric display. Type:

```
CAT   (Return), (EXECUTE), or (EXEC)
```

to list the directory of the *Manual Examples Disc* on the alpha screen.

To show both alpha and graphics displays simultaneously, press (GRAPHICS) or type:

```
GRAPHICS ON (Return), (EXECUTE), or (EXEC)
```

to bring back the graphics display. Now, remove the alpha display by pressing (GRAPHICS) again or typing:

```
ALPHA OFF (Return), (EXECUTE), or (EXEC)
```

Then press any key to superimpose the alpha display over the graph again.

Finish up by pressing (Clear display), (CLR SCR), or (CLR S) to clear the alpha display, and typing:

```
GCLEAR (Return), (EXECUTE), or (EXEC)
```

to clear the graphics display.

Press **EXIT** to stop the program.

Display-Enhancement Characters

Note

The Model 237 computer does not have blinking mode.

Display-enhancement characters are special, non-printable characters that affect how regular printable characters are displayed. Because these characters are non-printable, they are entered into the computer via their character codes.

A **character code** is simply a numeric code that represents a character to the computer. Since computers can store only numbers in their memories, character codes are used in lieu of actual characters.

The character code and display effect for each display-enhancement character is given in the following table.

Character Code	Action Resulting from Displaying the Character
128	All enhancements off (except colors on Model 236 with color CRT)
129	Inverse mode on.
130	Blinking mode on.
131	Inverse and Blinking modes on.
132	Underline mode on.
133	Underline and Inverse modes on.
134	Underline and Blinking modes on.
135	Underline, Inverse, and Blinking modes on.
136-143	Color enhancements on Model 236 with color CRT.

Do You Have Display Enhancements?

Your computer may or may not have display enhancements. To check if your computer has the alpha enhancements, type:

```
SYSTEM$("CRT ID")
```

(Return), (EXECUTE), or (EXEC)

Your computer will display a response similar to the following in the message/results line:

```
6:128HCGB
```

The "H" in the response indicates that the computer has "Highlights" (or display enhancements). If your computer's response has an "H", your computer has display enhancements; if your response does not have an "H", you do not have enhancements.

Incidentally, the number following the colon indicates the width of your display in characters. The "C" indicates Color capability; the "G" indicates Graphics; and the "B" indicates Bit-mapped display (Model 237). If any letter is missing, your computer does not have that particular capability.

Using Display Enhancements

Display-enhancement characters are entered into your computer with the CHR$ function. The CHR$ function takes a character code and converts it into the corresponding character.

Try it first with a printable character. Type:

```
PRINT CHR$(65)
```
(Return), (EXECUTE), or (EXEC)

As you can see, 65 is the character code for the letter "A". Unlike printable characters like "A", when you enter a display-enhancement character with the CHR$ function, you will not see a character displayed on the screen. What you *will* see is a change in all subsequent *printable* characters that are displayed.

Try It Yourself

This demonstration shows the effect of each display-enhancement character. First, display a normal line of text. Type:

```
PRINT "NORMAL"
```
Return, EXECUTE, or EXEC

Now, use the inverse mode (character code 129). Type:

```
PRINT CHR$(129);"INVERSE MODE"
```
Return, EXECUTE, or EXEC

A display enhancement character affects all subsequent printable characters until it is replaced by another display-enhancement character. Thus, inverse mode was "turned on" in the first part of the PRINT statement, and caused the following characters to be displayed in inverse mode. Now, replace inverse mode with blinking mode (character code 130). Press the **Recall**, RECALL, or RCL key and edit the line to read:

```
PRINT CHR$(130);"BLINKING MODE"
```
Return, EXECUTE, or EXEC

Next, change to inverse and blinking mode (character code 131):

```
PRINT CHR$(131);"INVERSE AND BLINKING MODE"
```
Return, EXECUTE, or EXEC

Next, change to underline mode (character code 132):

```
PRINT CHR$(132);"UNDERLINE MODE"
```
Return, EXECUTE, or EXEC

Next, change to underline and inverse mode (character code 133):

```
PRINT CHR$(133);"UNDERLINE AND INVERSE MODE"
```
Return, EXECUTE, EXEC

The underline is difficult to see, but it's there. Now, change to underline and blinking mode (character code 134):

```
PRINT CHR$(134);"UNDERLINE AND BLINKING MODE"
```
Return, EXECUTE, or EXEC

Now, put them *all* together in underline, inverse and blinking mode (character code 135):

```
PRINT CHR$(135);"UNDERLINE, INVERSE & BLINKING MODE"
```

Return, EXECUTE, or EXEC

Finally, turn off the display enhancements with the enhancements-off character (character code 128):

```
PRINT CHR$(128);"BACK TO NORMAL"
```

Return, EXECUTE, or EXEC

Be sure you remember to turn off the enhancements this way; they will stay on until you turn them off. Press Clear display, CLR SCR, or CLR S to clear the screen.

For more information on using the display enhancements, refer to *BASIC 3.0 Interfacing Techniques*, Chapter 8.

Graphics

In this section, you will run a few graphics demonstrations from the *Manual Examples Disc*. For details on how to do your own graphics programming, consult *BASIC 3.0 Programming Techniques* and *BASIC 3.0 Graphics Programming Techniques*.

Demonstrations

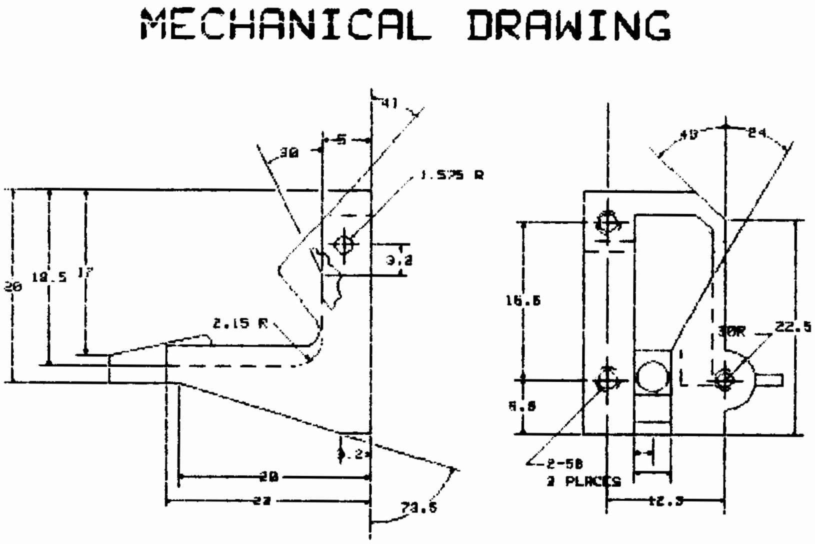
Insert the *Manual Examples Disc* into your default disc drive and close the door.

Now you're ready to load and run some demonstration programs. To get the first one, type:

```
LOAD "MECH",1
```
(Return), (EXECUTE), or (EXEC)

Placing the 1 after the program file name tells the computer to automatically begin executing the program after it has loaded it.

The following graphics display now appears on the screen. Press the softkey labeled **EXIT** ((f8) or (k8)) to clear the screen.



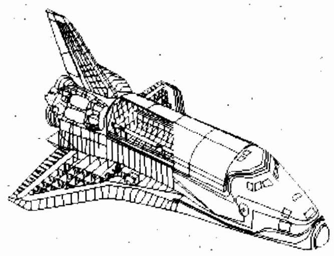
Mechanical Drawing Display

Now type the following line:

```
LOAD "SHUTTLE",1
```

(Return), (EXECUTE), or (EXEC)

Press (Return), (ENTER), or (ENTER) and **Continue**, (CONTINUE), or (CONT) as directed by the program to get a picture of the space shuttle.



Shuttle Display

Press (Return), (EXECUTE), or (EXEC) to exit the program.

If you're interested in seeing the BASIC program that draws the shuttle, type:

```
EDIT
```

(Return), (EXECUTE), or (EXEC)

Use the cursor control wheel, "mouse", or arrow keys to scroll through the program. Press unshifted-(Stop), (PAUSE), or (PSE) when you're finished.

Calculating While a Program Is Running

You can use your computer as a calculator even when the computer is busy running a program. Try this demonstration:

Try It Yourself

First, type the following line to clear any programs out of memory:

`SCRATCH` Return, EXECUTE, or EXEC

Now type:

`EDIT` Return, EXECUTE, or EXEC

to enter edit mode. Now enter the following six-line program that counts to ten forever. Press Return, ENTER, or ENTER after typing each line:

```
10  J=10
20  FOR I=1 TO J
30  PRINT I
40  NEXT I
50  GOTO 20
60  END
```

Now press **RUN** (unshifted-*f3*), RUN, or RUN to run the program. The program counts too fast for you to read the display, so press unshifted-Stop, PAUSE, or PSE to pause the program for a moment and verify that it is counting to 10 over and over again. Press **Step** (unshifted-*f1*), STEP, or STEP several times. Each time you press it, the program line displayed at the bottom of the screen is executed.

Now, press **Continue** (unshifted-*f2*), CONTINUE, or CONT) to resume program execution. Type in the following arithmetic problem:

`25+15` Return, EXECUTE, or EXEC

The computer delivers the answer, 40. Meanwhile, the program continues to run. Leave it running, and go on to the next section.

Changing Program Variables

You can interactively change program variables from the keyboard. The program you're running now counts to ten because the variable J is set to 10 in line 10. If you want the program to count from 1 to 20, just change the value of J by typing:

`J=20` (Return), (EXECUTE), or (EXEC)

Now press unshifted-(Stop), (PAUSE), or (PSE) again to pause the program. Notice that the program is now indeed counting from 1 to 20. Finally, press (Shift)-(Stop), (STOP) or (STOP) to terminate the program.

Configuring Your BASIC System

Chapter 8

Introduction

What does "configuring your BASIC system" mean? It simply means setting up the BASIC language system for your particular installation and needs. This chapter gives you the information and procedures you need to "customize" BASIC according to your requirements.

You received several discs with your BASIC 3.0 system. You will use these three BASIC discs to configure your system:

System Disc
Drivers Disc
Language Extensions Disc

The *System Disc* contains the core BASIC 3.0 language system, called `SYSTEM_BA3`.

The *Drivers Disc* contains several BIN files. Each BIN file is a driver that is required if you have a particular interface or product. For example, if you have an HP 98255 EPROM interface card, you must have BIN file EPROM loaded in order to use that interface.

The *Language Extensions Disc* also contains several BIN files. Each language extensions BIN file provides BASIC keywords or statements not in core BASIC. You may want to load one or more of these BIN files to extend your BASIC language capability. For example, if you have graphics capability, you will want to load BIN file GRAPH, and maybe GRAPHX.

The reason that all BIN files are not part of core BASIC or automatically loaded is that they take up computer memory. By configuring your own system and loading only the BIN files you will use, you conserve computer memory for other purposes.

You will perform the following steps as you work through this chapter. When you're finished, you will have configured BASIC just the way you want it.

1. Determine the BIN files your system requires.
2. Do some preliminary set-up procedures.
3. Load the driver BIN files.
4. Load the language extension BIN files.
5. Decide how you want to configure your system and do the configuration procedure you decided on.

It's now time to decide what BIN files you'll need for your particular system.

1. Choose Your BIN Files

The first task in configuring your system is to decide what BIN files you need for your system and for your programming requirements. The following tables list all the available BIN files.

The first table lists the BIN files that are on the *Drivers Disc.* The second table lists the BIN files that are on the *Language Extensions Disc.*

Read the definition of each BIN file and **circle** those BIN files you need to load.

Driver BIN Files

The following BIN files reside on the *Driver's Disc.*

File Name	Definition
BCD	Required for the HP 98623 Binary Coded Decimal interface.
BUBBLE	Required for the HP 98259 Magnetic Memory interface.
CRTA	Required for non-bit-mapped display. This BIN file is also part of SYSTEM_BA3 on the *System Disc* and is automatically loaded when the BASIC system is booted.
CRTB	Required for bit-mapped display. This BIN file is also part of SYSTEM_BA3 on the *System Disc* and is automatically loaded when the BASIC system is booted.
CS80	Required for CS80 disc drives. HPIB or FHPIB also required.
DCOMM	Required for the HP 98628 Datacomm or HP 98629 Shared Resource Management interface. If it's for the SRM interface, the SRM BIN file is also required. If loading **from** SRM, the SRM BIN file must be loaded before DCOMM.
DISC	Required for non-CS80 external disc drives. HPIB BIN file also required for HP 82901, HP 82902, HP 8290X, and HP 9135 drives. HPIB or FHPIB required for other disc drives.
EPROM	Required for the HP 98255 Erasable Programmable Read-Only Memory interface.
FHPIB	Required for the HP 98625 High-speed Disc interface. Achieves higher disc drive performance than HPIB BIN file when used with DISC or CS80 BIN files.
GPIO	Required for the HP 98622 General Purpose Input/Output interface.
HP9885	Required for the HP 9885 Disc Drive. The drive is connected to the HP 98622 GPIO interface.
HPIB	Required for the internal (built-in) HP-IB interface and for the HP 98624 HP-IB interface.
SERIAL	Required for the HP 98626 Asynchronous Serial Interface.

Language Extensions BIN Files

The following BIN files reside on the *Language Extensions Disc*:

File Name Definition

CLOCK The TIMEDATE function is in core BASIC. CLOCK provides these functions and statements:

DATE
DATE$
ON CYCLE
OFF CYCLE
ON DELAY
OFF DELAY
ON TIME
OFF TIME
TIME
TIME$

GRAPH Provides the RATIO function and these commonly used graphics statements:

ALPHA ON
ALPHA OFF
AXES
CLIP
CSIZE
DRAW
DUMP DEVICE IS
DUMP GRAPHICS
FRAME
GCLEAR
GINIT
GLOAD
GRAPHICS ON
GRAPHICS OFF
GRID
GSTORE
IDRAW
IMOVE
IPLOT
LABEL
LDIR
LINE TYPE
LORG
MOVE
PEN
PENUP
PDIR
PIVOT
PLOT
PLOTTER IS
RPLOT
SHOW
VIEWPORT
WINDOW

ERR Provides text messages for errors.

GRAPHX Provides these more advanced graphics statements for graphics input, color plotting, and area filling:

AREA COLOR
AREA INTENSITY
AREA PEN
DIGITIZE
GESCAPE
GRAPHICS INPUT IS
IPLOT (*)
PLOT (*)
POLYGON
POLYLINE
READ LOCATOR
RECTANGLE
RPLOT (*)
SET ECHO
SET LOCATOR
SET PEN COLOR
SET PEN INTENSITY
SYMBOL
TRACK IS ON
TRACK IS OFF
WHERE

IO Provides the PPOLL and SPOLL functions and these statements used to interface with IO resources:

ABORT
BREAK
CLEAR
DISABLE INTR
ENABLE INTR
LOCAL
LOCAL LOCKOUT
ON INTR
OFF INTR
ON SIGNAL
OFF SIGNAL
PASS CONTROL
PPOLL CONFIGURE
PPOLL RESPONSE
PPOLL UNCONFIGURE
REMOTE
REQUEST
RESET
SEND
SIGNAL
TRIGGER

KBD Provides these statements for advanced use of the keyboard, including softkeys. KBD is required to use the HP 46060A "mouse" with the HP 46020A keyboard):

EDIT KEY
LIST KEY
LOAD KEY
RE-STORE KEY
SCRATCH KEY
STORE KEY

LEX Provides this statement for changing collate sequences, and is required to convert keycodes to characters for local language keyboards:

LEXICAL ORDER IS

MAT Provides these statements and functions which let you handle arrays, or matrices:

BASE
DET
DOT
MAT
MAT REORDER
MAT SORT
MAX
MIN
RANK
REDIM
SIZE
SUM

MS Extends the CAT statement. It also provides these additional mass memory statements:

CHECKREAD
PRINT LABEL
READ LABEL

PDEV Provides these commands and statements which are useful for writing and debugging programs:

CHANGE
COPYLINES
FIND
INDENT
LOADSUB FROM
MOVELINES
SECURE
TRACE ALL
TRACE OFF
TRACE PAUSE

SRM Required for the HP 98629 Shared Resource Manager interface. DCOMM driver also required. SRM must be loaded before DCOMM **if** loading from SRM network. SRM also provides an alternate form of the PROTECT statement appropriate for the SRM. Loading SRM sets the system clock on computers that do not have the powerfail, real-time clock feature. Provides these statements:

CREATE DIR
LOCK
UNLOCK

TRANS Enables use of buffers and provides these statements which support the TRANSFER statement:

ABORTIO
ON EOR
OFF EOR
ON EOT
OFF EOT
TRANSFER
WAIT FOR EOR
WAIT FOR EOT

XREF Provides the XREF command which obtains a cross-reference listing of identifiers in a program or subprogram.

KNB2_0 When loaded, this program causes KNOBX to function as it did in BASIC 2.0/2.1. Do not load this BIN file until you fully understand its purpose and operation. Refer to the Knob section in Chapter 15 of the *BASIC 3.0 Programming Techniques* manual for more information.

2. Prepare to Load

You now know which BIN files you need to load to successfully configure your BASIC system. Now let's determine what BIN files you have in memory.

You previously booted the BASIC system from the *System Disc* (Chapter 2). Then you loaded BIN file ERR. If you booted from an external disc drive, you also loaded DISC, HPIB, and CS80 from the *Drivers Disc.* This allowed your computer to "talk" to its disc drive(s) so you could perform the procedures in the guide.

Now type:

`LIST BIN` (Return), (EXECUTE), or (EXEC)

to see what BIN files you have in computer memory. If you have worked straight through this guide, you should have as least the following:

ERR
CRTB
CRTA

If you booted from an external disc, you should also have:

DISC
CS80
HPIB

If, however, you just powered up with your *System Disc*, you will only have:

CRTB
CRTA

Now type:

`SCRATCH BIN` (Return), (EXECUTE), or (EXEC)

This deletes all BIN files except the CRTA or CRTB file required for your display.

3. Load Driver BIN Files

1. Insert the *Drivers Disc* into your default msus disc drive.
2. For each driver BIN file that you circled in the preceding Driver BIN Files table, type the following:

   `LOAD BIN "`**filename**`"` (Return), (EXECUTE), or (EXEC)

   where **filename** is the name of the BIN file you circled.

Note

Remember to load the *other* CRT BIN file if **both** display types will be used with your BASIC system.

3. Remove the *Driver's Disc* from your drive.

4. Load Language Extensions BIN Files

1. Insert the *Language Extensions Disc* into your default msus disc drive.
2. For each language extensions BIN file that you circled in the preceding Language Extensions BIN Files table, type the following:

   `LOAD BIN "`**filename**`"` (Return), (EXECUTE), or (EXEC)

   where **filename** is the name of the BIN file you circled.
3. Remove the *Language Extensions Disc* from your drive.

5. Configure Your System

There are three basic ways you can configure your system:

- Load the required BIN files one at a time each time you boot the BASIC system. (You've just done this in the preceding steps.)
- Use the STORE SYSTEM command to create a file that contains the entire configured BASIC system.
- Write an autostart program which will automatically load predetermined BIN files when BASIC is booted.

The following instructions explain the three methods.

Hand-loading BIN Files

If you choose this method, each time you boot BASIC you must load the required BIN files one at a time from the flexible discs using the LOAD BIN statement. This is the method you used when you loaded BIN files in previous steps.

Creating a System File

This method is particularly good if you would like to configure more than one system. You can store the system file on either hard or flexible disc, but be aware of disc size limitations.

1. Ensure that you have all the required BIN files loaded into computer memory.
2. Type the following:

   `STORE SYSTEM` "**filename:msus**" (Return) (EXECUTE), or (EXEC)

   where **filename** is the name of your system file; and
   where **msus** is the mass storage unit specifier of your storage device

   For example:

   ```
   STORE SYSTEM "SYSTEM_IO:,700"
   ```

   stores your configured BASIC system (core BASIC and all BIN files loaded in computer memory) into a file named SYSTEM_IO at the mass storage device at select code 7, primary address 0.

Note

You must use SYSTEM_ as the first seven characters of your system file name if you have a boot ROM earlier than 3.0. If you have boot ROM 3.0 or later you can use SYSTEM_ or SYS as the file name prefix (e.g., SYSTEM_IO or SYSGRAPH).

3. Repeat the above procedure for each unique system you want to configure. To configure a new system:
   a. Execute SCRATCH BIN to delete all BIN files except the CRT BIN for your display.
   b. Load the new set of required BIN files.
   c. Store the new system using a unique file name (e.g., SYSTEM_GRA).

To access the SYSTM files you have created, re-boot your system and press the space bar several times.

If you have boot ROM 3.0 or later, all available systems are displayed. Type the two characters displayed just to the left of the system you wish to select. Your computer will boot the configured system that you have selected.

If you have an earlier boot ROM (1.0 or 2.0), the computer boots the first SYSTM file it finds that begins with SYSTEM_. If you want a different system file booted, rename all other system files to eliminate the SYSTEM_ prefix. For example, assume you have three system files named SYSTEM_BA1, SYSTEM_BA2, and SYSTEM_BA3. To be sure you load SYSTEM_BA3, rename the others to BA1 and BA2.

Using an Autostart Program

You may want to use this method if you want to configure just one system and you do not have a hard disc.

1. Ensure that you have all the required BIN files loaded into computer memory.
2. Write the following program:

```
10 ! Autostart of a larger configuration program.
20 PRINT "Remove BASIC 3.0 System Disc."
30 PRINT "Insert disc containing configuration program."
40 INPUT "Press Return or ENTER when ready.",C$
50 LOAD "AUTOPROG"
60 END
```

3. Ensure that the *System Disc* is in the flexible disc drive from which you booted or the default disc drive.
4. Type:

   `STORE "AUTOST"` (Return), (EXECUTE), or (EXEC)

   This stores the autostart program on the *System Disc* as a PROG file. It is automatically executed when BASIC is booted from that disc.

5. Create a program called AUTOPROG and store it on another flexible disc. This program will actually configure your system. Here is an **example**:

```
 10 PRINT "Insert BASIC 3.0 Drivers Disc."
 20 INPUT "Press Return or ENTER when ready.",C$
 30 PRINT "Please wait... Loading driver BIN files."
 40 LOAD BIN "DISC"
 50 LOAD BIN "HPIB"
 60 LOAD BIN "IO"
 70 LOAD BIN "GPIO"
 80 PRINT "Remove BASIC 3.0 Drivers Disc."
 90 PRINT "Insert BASIC 3.0 Language Extensions Disc."
100 INPUT "Press Return or ENTER when ready.",C$
110 PRINT "Please wait... Loading language extension BIN files."
120 LOAD BIN "GRAPH"
130 LOAD BIN "ERR"
140 LOAD BIN "KBD"
150 PRINT "BASIC configured"
160 PRINT "Remove BASIC 3.0 Language Extensions Disc."
170  END
```

Store your program on the disc you intend to use for configuration by executing a `STORE "AUTOPROG"` statement.

Upgrading from Previous Versions

If you are upgrading to BASIC 3.0 from BASIC 2.0 or 2.1, you need additional information. Refer to *BASIC 3.0 Programming Techniques*, Chapter 15.

Task Reference

Appendix

A

This appendix provides quick-reference, task-oriented procedures. If you have worked through this guide or are familiar with BASIC programming, you should be comfortable with this section.

All the tasks described in detail in this guide are given here in condensed form. This appendix should serve to jog your memory when you want to perform a certain task.

Note

When a keypress is required, all three keyboard keys are given in this order: HP 46020A, HP 98203B, and HP 98203A.

The word "execute" in this appendix simply means to press the appropriate key on your keyboard:
Return, EXECUTE, or EXEC, depending on your keyboard.

The index for this appendix follows:

ASCII Characters

To find an ASCII character from its decimal equivalent:

Press: **Any char**, (ANY CHAR), or (ANY C)

Type the three-digit decimal equivalent of the character you want displayed.

The character is displayed.

Example:
Press: **Any char**, (ANY CHAR), or (ANY C)

Type: `065`

The letter `A` is displayed.

Arithmetic Hierarchy

Your computer evaluates mathematical expressions in the following order:

1. Parentheses (if nested, inner-most parentheses evaluated first)
2. Exponentiation
3. Multiplication/division (left to right in order)
4. Addition/subtraction (left to right in order)

Example:

```
2+3*32/((7-6)+3)^2
2+3*32/(1+3)^2
2+3*32/4^2
2+3*32/16
2+96/16
2+6
8
```

Autostart File

Write an autostart program and store it by executing:

```
STORE "AUTOST"
```

This stores the autostart program on the system disc as a PROG file. It is automatically executed when the system is booted.

BIN Files

Deleting

To delete all BIN files from computer memory except the BIN file required for your display, execute:

```
SCRATCH BIN
```

Note

You cannot delete individual BIN files.

Listing

To see what BIN files are loaded in computer memory, execute:

```
LIST BIN
```

All BIN files loaded in memory are displayed.

Loading

To load a BIN file from a disc, execute:

```
LOAD BIN "name:msus"
```

where **name** is the BIN file name, and **msus** is the mass storage unit specifier of the disc drive

The msus is unnecessary if the disc drive is specified as the default mass storage device (refer to msus in this appendix).

Example:

```
LOAD BIN "GRAPH:,701"
LOAD BIN "GRAPH"
```

Boot ROM

To determine the revision of your boot ROM, turn on your computer without a system connected.

If you have a boot ROM 3.0 or later, the revision is displayed.

If `UNABLE TO FIND SYSTEM` appears at the bottom of the display, you have a boot ROM of revision 1.0 or 2.0.

Booting BASIC

With Boot ROMs 1.0 or 2.0

If BASIC is the only system "on line", it is automatically booted at powerup.

If more than one system is "on line", the boot ROM lists the first letter of each system and waits 10 seconds for a key press to select a system. If no key is pressed, the first system found is booted.

Example:

```
WHICH SYSTEM?
B P H
```

In this example, the computer found three systems: BASIC, Pascal, and HPL. If none of the three keys is pressed within 10 seconds, the computer loads BASIC.

With Boot ROMs 3.0 and Later

The computer goes through a pre-determined search sequence at powerup and boots the first "on line" system it finds.

If more than one system is "on line", you **can** choose the system you want booted by pressing the space bar a few times while the computer is running its self-test. The computer responds by listing all "on line" systems preceded by two-character codes. Type the two-character code for the system you want booted.

Example:

```
:HP9895, 700, 0
   1P SYSPASCAL
:HP8290X, 702, 0
   1B SYSTEM_BA3
```

Type: `1B` to select BASIC

CRT

Configuration

To determine your CRT's configuration, execute:

```
SYSTEM$("CRT ID")
```

Your computer displays a response similar to this:

```
6:128HCGB
```

The number following the colon (`:`) specifies your display's character width (128 in the example).

The `H` specifies that your CRT has display enhancements; the absence of `H` means no display enhancements.

The `C` means your display has color capability; its absence means you have a monochrome display.

The `G` indicates graphics capability; you do not have graphics if no `G` appears.

The `B` means you have a bit-mapped display; otherwise, you have a non-bit-mapped display.

Hz Setting

To change the Hz setting of your display from 50 Hz to 60 Hz or vice versa, press the CTRL and C keys at the same time during self-testing or booting.

In response to the display, press either 5 to select 50 Hz or 6 to select 60 Hz.

Discs

Copying Flexible Discs with Two Drives

1. Insert source disc into one drive and initialized destination disc in the other drive.
2. Execute:

   `COPY ":`**source msus**`" TO ":`**destination msus**`"`

   where the **source msus** is the mass storage unit specifier of the drive containing the disc you want to copy, and **destination msus** is the mass storage unit specifier of the drive containing the initialized destination disc

Copying Flexible Discs with One Drive

1. Execute: `INITIALIZE ":MEMORY,0"`
2. Insert source disc into disc drive.
3. Execute: `COPY ":`**msus**`" TO ":MEMORY,0"`

   where **msus** is the mass storage unit specifier of the disc drive
4. When copying is done, remove the source disc from the disc drive and insert an initialized destination disc into the drive.
5. Execute: `COPY ":MEMORY,0" TO ":`**msus**`"`

   where **msus** is the mass storage unit specifier of the disc drive

Initializing

CAUTION

INITIALIZING A DISC DESTROYS DATA OR PROGRAMS RECORDED ON THE MEDIA. BE SURE YOU KNOW WHAT DISC YOU ARE USING. YOU CAN INADVERTENTLY WIPE OUT YOUR BASIC 3.0 LANGUAGE SYSTEM, OR OTHER FILES, IF YOU INITIALIZE THE WRONG DISC.

IF YOU INADVERTENTLY INITIALIZE A HARD DISC DRIVE, DO NOT INTERRUPT THE INITIALIZATION PROCESS. WAIT UNTIL THE DISC DRIVE LIGHT GOES OFF BEFORE CONTINUING. THIS CAN TAKE OVER AN HOUR. INTERRUPTING A HARD DISC DRIVE INITIALIZATION CAN SEVERELY DAMAGE THE DRIVE.

Some disc drives provide disc formatting options that are selected with the INITIALIZE statement. For example, you can select single-sided or double-sided when initializing a disc on the HP 9122 disc drive. Refer to your disc drive manual for details.

1. If you're initializing a flexible disc, insert the disc to be initialized into a flexible disc drive.
2. Execute: `INITIALIZE ":`**msus**`"`

   where **msus** is the mass storage unit specifier of the disc drive

Note

INITIALIZE, unlike other statements, **requires** the msus. This helps prevent inadvertent destruction of data by initializing the wrong disc.

Listing a Directory

To list the disc's directory, or catalog, execute:

```
CAT ":msus"
```

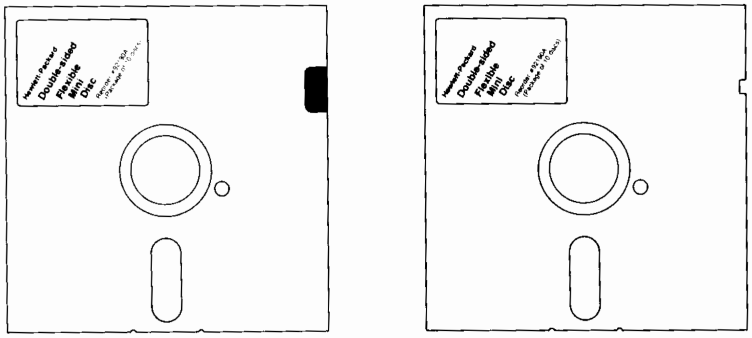
where **msus** is the mass storage unit specifier of the disc drive containing the disc you want to catalog

The msus is unnecessary if the disc drive is specified as the default mass storage device (refer to msus in the appendix).

Example:

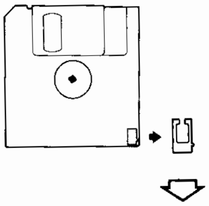
```
CAT ":,700,2"
CAT
```

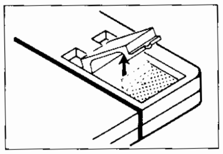
Write-protecting Flexible Discs



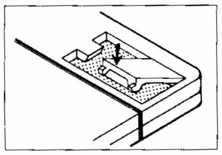
Write-protected **Write-enabled**

5¼-inch Mini-discs

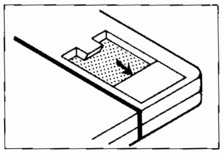




1. Weaken or score the attach point so as not to break the tab.
   Break off the write-protect tab.

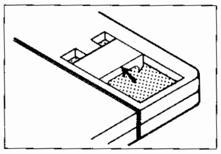


2. Align the protrusion on the tab with the groove in the disc.



Write-protect

3. Depress the tab into the groove–tab should fit snugly. In the position shown, the tab write-protects the disc.



Write-enable

4. To write-enable the disc, slide the tab up.

3½-inch Micro-discs

Display Enhancements

To determine if your computer has display enhancements, execute:

```
SYSTEM$("CRT ID")
```

If an H is included in the displayed response, you have display enhancements; if not, you don't.

Select display enhancements with the CHR$ function according to the following table. Note that the Model 237 does not have blinking mode.

Character Code	Action Resulting from Displaying the Character
128	All enhancements off (except colors on Model 236 with color CRT).
129	Inverse mode on.
130	Blinking mode on.
131	Inverse and Blinking modes on.
132	Underline mode on.
133	Underline and Inverse modes on.
134	Underline and Blinking modes on.
135	Underline, Inverse, and Blinking modes on.
136-143	Color enhancements on Model 236 with color CRT.

Files, Copying

To copy one file from a disc to another disc, execute:

```
COPY "oldname:source msus" TO "newname:dest msus"
```

where **oldname** is the name of the file you are copying; **source msus** is the msus of the disc drive you're copying from; **newname** is the name of the duplicate file you're creating (can be same as **oldname**); and **dest msus** is the msus of the disc drive you're copying to.

Example:

```
COPY "WORK:,700" TO "WORK:,701"
```

The file named WORK is copied from a disc at address 700 to a disc at 701; the new file is also named WORK.

msus

Determining an msus

You can determine an msus in one of two ways:

- If you're determining the msus of a flexible disc drive, take all systems "off line". Insert your BASIC system disc into the drive and power up. The msus is listed in the upper right corner of the display.
- Refer to the table on page 103.

Setting a Default msus

To automatically refer all mass storage operations to a specific drive, execute:

```
MSI "msus"
```

where **msus** is the msus of the default drive. MSI is an abbreviation for MASS STORAGE IS.

Example:

```
MSI ":,701"
```

could set the flexible disc drive of the HP 9135 as the default disc drive. All mass storage operations would default to that drive unless another msus was specified in the statement.

Printer

Setting Hard-copy Printer as Default Device

To specify a hard-copy printer as the default printer in your system, execute:

```
PRINTER IS PRT
```

if you have just one printer in your system; or

`PRINTER IS` **device selector**

if you have more than one printer.

where **device selector** is the device selector of the printer.

Example:

```
PRINTER IS 701
```

selects the printer connected to the internal HP-IB (select code 7) that is set to primary address 1.

Setting CRT as Default Printer

To set the CRT as the default system printer, execute:

```
PRINTER IS CRT
```

Note that the CRT is automatically selected as the default printer at powerup.

Dumping to a Printer

To dump displays to a printer, use the `DUMP ALPHA` and `DUMP GRAPHICS` statements.

If you want to dump to a different printer than the default printer, execute:

`DUMP DEVICE IS` **device selector**

or

`DUMP ALPHA` **#device selector**
`DUMP GRAPHICS` **#device selector**

where **device selector** is the device selector of the printer you want to dump the display to.

Programs

Clearing Memory

To clear computer memory in preparation for writing a program, execute:

```
SCRATCH
```

Editing

To get into edit mode, either to begin writing a program or to edit an existing program, execute:

```
EDIT
```

Listing

To list a program, execute:

```
LIST
```

to list the program on the default printer; or

```
LIST #device selector
```

to list the program on the **device selector** printer if it is not the default printer

Example:

```
LIST #701
```

Loading

To load a program into computer memory from a mass storage device, execute:

```
LOAD "name:msus"
```

to load a PROG file;

> where **name** is the program name and **msus** is the device containing the program (msus is unnecessary if it is the default device).

```
GET "name:msus"
```

to load an ASCII file.

Running

To run a program, execute:

```
RUN
```

or press: **Run**, (RUN), or (RUN)

Your computer can perform calculations executed from the keyboard while a program is running.

You can execute changes in program variables at the keyboard while the program is running. The program continues to run with the new variable.

Storing

To store a program from computer memory to a mass storage device, execute:

```
STORE "name:msus"
```

to store a PROG file;

> where **name** is the program name and **msus** is the device containing the program (msus is unnecessary if it is the default device).

```
SAVE "name:msus"
```

to store an ASCII file.

Softkeys, HP 46020A

The (Menu) and (System) keys control softkeys on the HP 46020A keyboard.

(System) selects System softkey mode.

(User) selects User 1 mode.

When the keyboard is in a user mode, (Shift)-(Menu) cycles softkey menus through User modes (User 1 → User 2 → User 3 → User 1, etc.).

Unshifted-(Menu) toggles the softkey display on and off, irrespective of the softkey mode.

System, Configuring a

You can configure a BASIC system by:

- Hand-loading the system and all required BIN files on powerup.
- Loading the system and required BIN files into your computer memory and executing:

  STORE SYSTEM "**filename:msus**"

  where **filename** is the name of your system file; and where **msus** is the mass storage unit specifier of your storage device

  filename must have a prefix of SYSTEM_ if you have a 1.0 or 2.0 boot ROM; otherwise, it must have a prefix of SYSTEM_ or SYS
- Writing an autostart program that automatically loads predetermined BIN files when the discs are inserted. Store the autostart file on the *System Disc* by executing:

  ```
  STORE "AUTOST"
  ```

BASIC Documentation

Appendix **B**

As with most products, learning how to use the manuals properly will help you to get the most from the product. In order to use the manuals most effectively, you should know both the objective and content of each manual. This appendix lists the manual set and gives a brief description of each of the major manuals.

BASIC 3.0 Manuals

Part Number	Description
98613-90010	BASIC 3.0 Programming Techniques
98613-90020	BASIC 3.0 Interfacing Techniques
98613-90030	BASIC 3.0 Graphics Programming Techniques
98613-90040	BASIC 3.0 User's Guide
98613-90050	BASIC 3.0 Language Reference
98619-90051	SRM HP Series 200 Workstation Manual
98613-90070	BASIC 3.0 Master Index
98613-90081	BASIC 3.0 Software and Manual Catalog
98613-10010	CSUB Preparation Manual
98613-10020	BASIC 3.0 Utilities Library Manual
98613-10030	Loader Utility Manual
98613-87901	BASIC 3.0 Documentation Package

BASIC 3.0 Programming Techniques described writing, editing, storing, running and debugging BASIC programs. The manual also describes such programming topics as string and math operations, using the real-time clock, and communicating with the operator.

BASIC 3.0 Interfacing Techniques describes how to communicate with external devices. Both general and interface-specific techniques are described in the manual.

Read Chapter 1, "Manual Overview", to see this manual's objectives and contents. This chapter also describes the organization of information in the manual and briefly describes each chapter. You may want to scan chapters of interest in the main part of the manual. The "Useful Tables" contains information relevant to interfacing, and the Index provides an index to the topics in this manual.

BASIC 3.0 Graphics Programming Techniques describes using the graphics capabilities of Series 200 computers. Plotting on the CRT and on external graphics devices are fully described in this manual, as well as using external graphics input devices.

Chapter 1, "Introduction to Graphics", describes the objectives of the manual and assumptions made about your knowledge of BASIC programming. You may want to scan the individual chapters of the manual as your interest dictates. An index is also provided in this manual.

BASIC 3.0 User's Guide introduces you to BASIC on your Series 200 computer. It describes many of the operating features such as keyboards, displays, and discs. It show you how to do simple operations on the computer.

BASIC 3.0 Language Reference provides a complete "dictionary" of precise descriptions of every keyword in the Series 200 BASIC language. Drawings graphically show the proper syntax of each keyword, and any parameters are described in an accompanying table. The semantics section describes the resultant action of different keyword syntaxes.

The "Keyword Dictionary" section is the main part of this manual, providing the following three sections: 1) "Language History", which provides valuable information about how and when the language has been revised and updated; 2) "Using the Keyword Dictionary", which describes what information is provided by the dictionary and explains how to use it; and finally 3) the actual dictionary entries. You should read the first of these sections before attempting to use the rest of the manual.

The "Glossary" provides concise definitions of technical terms used throughout the manual set, which you can refer to as you encounter unfamiliar terms. The "Interface Registers" section contains listings of all status and control registers of I/O paths, CRT, keyboard, and optional interfaces. The contents of the "Useful Tables" and "Error Messages" sections are self-evident. The "Keyword Summary" section provides a complete list of keywords in the Series 200 BASIC language, grouped according to the function that it performs.

Glossary Of Terms

Alpha Display: The part of the screen defined by BASIC for displaying normal text characters (letters and numbers).

Binary Programs: Programs written in machine language which extend the command set and capabilities of a language system.

Bit: A binary digit (1 or 0).

Bit-mapped Display: A display with a graphics scheme which allows the user to select (turn on or off) a single pixel, or point, on the display. The Model 237 has a bit-mapped display.

Boot ROM: The Boot ROM stores instructions that tell the computer how to search for a system program. They basically keep the computer running until a system program can take over.

Bus Address: A number that identifies the location of a device on the HP-IB; also called primary address.

Byte: The unit of memory used on your computer. One byte equals eight bits.

Character Code: A numeric code which is used to represent a character inside the computer.

CRT: The computer's screen (cathode ray tube).

Cursor: The underline character that marks the position on the screen where the next character will be typed.

Default Device: If you execute an I/O operation (e.g., LOAD, PRINT) without specifying which device the operation refers to, the computer *assumes* you want to use the default device and carries out the command. When loading a program from the default disc drive, for example, you don't have to include the msus–the computer *assumes* the msus of the default drive.

Disc: Similar to a phonograph record, except that it stores programs and data instead of music.

Disc Drive: An input/output device that transfers programs and data between a disc and the computer's memory.

Graphics Display: The part of the screen defined by BASIC for drawing graphs, charts, and other pictorial displays.

Hardware: All of the electrical and mechanical components of the computer.

Human Interface: The part of a program that handles all interaction between the computer and the person who operates it. A good human interface makes using the computer easy.

Input/Output: Anything relating to the exchange of information between the computer and its peripherals.

Input Device: A peripheral device which transfers programs and data into the computer. Common input devices include keyboards, disc drives, and graphics tablets.

K bytes: 1 024 bytes.

Language System: A large program which performs all of the functions of a system program, plus supports a programming language like BASIC or Pascal.

Language-dependent Program: A program which requires a language system in order to run. Language-dependent programs are always loaded into memory *after* a language system has been booted.

Machine Language: The language the computer's processor understands, expressed in terms of bits (1s and 0s).

Mass Storage Device: For most applications, a mass storage device is simply a disc drive.

Mass Storage Unit Specifier: A string of characters used to designate a particular disc drive (or other mass storage device). Abbreviated as msus.

Memory: The area of the computer where programs and data are stored. The processor cannot run a program unless it is in memory.

Memory Address: A number which uniquely identifies one byte of memory.

Menu: Usually, a list of things that a program can do. You control the progress of the program by selecting one of the menu items.

Mouse: A hand-operated device that moves the cursor in all directions and scrolls the display. It also has programmer-definable buttons.

msus: See Mass Storage Unit Specifier.

Output Device: A peripheral device which accepts information from the computer for storage or display purposes. Common output devices include computer screens, disc drives, printers and plotters.

Peripheral Devices: Devices that allow the computer to communicate with the outside world. See "Input Device" and "Output Device."

Primary Address: See Bus Address.

Program: A set of instructions that tell the processor how to perform a particular task. Most programs are written in a high-level programming language like BASIC.

Program Listing: A list of all statements in a program.

Prompt: A message that a running program displays on the screen when it needs data or other information from you.

RAM: Random Access Memory. This is erasable program memory. Programs and data are usually copied into RAM from a disc drive or other mass storage device, executed by the processor, and then erased from RAM. When the power is turned off, RAM is erased.

ROM: Read-Only Memory. This is permanent program memory, used primarily for storing essential programs. Programs in ROM are never erased, so ROM is not reusable.

Select Code: A number which uniquely identifies an interface. The processor uses the select code to select which interface will be used in a data transfer operation.

Software: A synonym for program.

System Program: A program which handles all of the overhead functions of computing, such as defining the keyboard, managing the peripherals, refreshing the display, etc. When the computer is turned on and passes its self-test, it immediately begins searching for a system program to boot.

Stand-alone Program: A program that has a "built-in" system program and can run without any underlying language support.

Subject Index

HEWLETT
PACKARD

Part No. 98613-90040
E 1284
Microfiche No. 98613-99040

Printed in U.S.A.
First Edition with update merged
December 1984