HEWLETT-PACKARD

Software Development System Utilities Reference Manual





HP-94 Handheld Industrial Computer

Software Development System Utilities Reference Manual



Edition 1 December 1986 Reorder Number 82520-90003

Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

• Copyright 1986, Hewlett-Packard Company.

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

MSo-DOS is a registered trademark of Microsoft Corporation.

Portable Computer Division 1000 N.E. Circle Blvd. Corvallis, OR 97330 U.S.A.

Printing History

Edition 1

December 1986

Mfg. No. 82520-90004

Introduction

Introduction

The HP-94 Utilities Reference Manual provides programming reference material for the HP-94 Handheld Industrial Computer. You need some knowledge of the BASIC programming language and some programming experience to make the best use of this manual.

The manual consists of the following sections:

- The Introduction provides general information that applies to all of the Software Development System (SDS) keywords. It explains how to interpret parameter tables and syntax diagrams in the keyword dictionary and also contains information relating to the MS-DOS file structure.
- The HXBASIC Program Development Utility chapter defines the BASIC language programming environment and includes information on the File Management Commands, Program Editing Commands, and Program Debugging Commands.
- The HXC File Conversion Utility chapter tells you how to convert your files to a form that can be transferred to the HP-94.
- The HXCHRSET Roman-8 Utility chapter tells you how to select the Roman-8 or the standard character set on your development-system computer.
- The **HXCOPY File Copy Utility** chapter tells you how to transfer programs and files between your development-system computer and the HP-94.
- The HXMODE Handshaking Utility chapter tells you how to provide XON/XOFF handshaking between your development-system computer and the HP-94.
- The HP-94 Operating-System Commands chapter describes the file-management functions and limited self-test capabilities in the HP-94.
- The SYBD HP-94 BASIC Debugger chapter tells you how to do final program debugging in the HP-94.
- **Appendix A Error Handling** describes errors and what to do about them.
- Appendix B Diagnostic Tests describes the tests you can run to verify the functionality of the HP-94.

Keyword Descriptions

Each keyword is defined by a description of the keyword, a syntax diagram showing pictorially how the keyword is used, and a table listing parameters and their allowable ranges. Examples of the use of

the keyword and related keywords are listed.

The Syntax Diagram

The syntax diagram shows pictorially how to assemble a proper expression, statement, or command using the keyword. Items enclosed in ovals, circles, and rectangles are the elements of the expressions, statements, and commands.

Format Conventions: The following syntax-diagram format conventions are used in this manual:

- The elements enclosed in ovals are keywords that must be typed in exactly as shown, except that uppercase and lowercase letters may be used interchangeably.
- The elements enclosed in circles are punctuation or keys that must be typed in exactly as shown.
- The elements enclosed in rectangles are parameters which are described in the table. Generally, uppercase and lowercase letters are *not* interchangeable.
- The elements are connected into paths by arrows. Starting at the left of the diagram, you may follow any path in the direction indicated by the associated arrows. You must, however, end at the far right of the diagram.

If several paths exist around one or more elements, each of the paths is optional; you should follow the path that does what you want to do. For example, LIST, LIST 10, and LIST 10, 100 are all valid commands. Many optional elements have default values listed in the table of parameters.

Line numbers and line labels are not shown in the syntax diagrams.



Space Conventions: The following conventions for the use of spaces are used in this manual:

- You may use one or more spaces between elements shown connected by an arrow.
- Consecutive ovals *must* have at least one space separating them.
- You may not use spaces between elements shown next to each other on a path without an arrow connecting them.

Table of Parameters

The table describes each parameter used in the syntax diagram. When the parameter is required to

assume values within a specified range, that range is listed. A dash ("-") indicates no range restrictions.

Understanding Files

The following paragraphs contain the basic information you should know concerning MS-DOS files and filenames. BASIC creates its own environment within the MS-DOS file structure. This environment includes certain file types and file security. BASIC-type files can be created, accessed, copied, and purged within BASIC.

Data and Program Files: A file is a collection of information stored on a disc under a filename.

A program is a set of instructions in a programming language, such as BASIC, which tells the computer what to do.

Programs are stored in files called program files. When you want to run a program, you use the filename to tell the computer which program you want.

Most application programs store data in data files.

As part of the filename, you can use an optional three-character extension to identify similar files or to describe a file more completely.

Files may be saved on discs in directories or as non-directory files. Directories are files that contain other files. They provide a convenient way to group files of similar types. Every disc has at least one directory, the Root directory. In addition to the Root directory, you can create your own directories and subdirectories (directories contained within a directory). The list in sequence of directories and subdirectories the computer has to go through to locate a particular file plus the name of the file itself including the three-character extension, if any, is the pathname for that file.

MS-DOS commands are not concerned with the content of the files but operate on the files themselves. The MS-DOS file structure allows data and program files to be treated almost identically. Therefore, in the remainder of this manual, a file may be either a data file or a program file.

File-Name Convention: The File Name is defined as *< filename > [. < extension >]* where the extension consists of three alphanumeric characters.

Use of the extension in naming a file is optional ([]); however, if you use an extension, or a file name has been created that includes an extension, the syntax diagram will show whether the extension must be used as part of the file name.

The Pathname for a particular file is defined as follows:

<device name>:<directory name><file name>

(There may be more than one directory name in the pathname.)

File Structure: MS-DOS uses a hierarchical file structure in the form of an upside down tree as shown below. You can go to another file in the directory in which you are working by simply typing in the name of the file. If you wish to specify a file in another directory, you will probably want to start back at the root directory and specify the full pathname to the file.



For example, your disc is in drive A; you turn on your computer, and you wish to specify File4. You type

A:\Data\Info\File4

You finish with File4 and now wish to specify the File3 file. You type

File3

Now you wish to specify File2. You type

File2

(You do not need to specify the device name again.)

For further details, consult the documentation for the computer you are using.

1-4 Introduction

Reserved File Extensions: The following file extensions are created by the utilities. You should avoid their use as files you create using them may be overwritten by the utilities.

BAS is created by the SAVE command in HXBASIC. REF is created by the XREF command in HXBASIC. LST is created by the LIST command in HXBASIC. MDS, MBK, MMP, BMP, CMD, and CBK are created by HXC.

File Security: You may secure BASIC program files by using the SECURE option of the SAVE command in HXBASIC.

HXBASIC Program Development Utility

HXBASIC Program Development Utility

The HXBASIC program development utility provides an environment for developing BASIC language programs for the HP-94 Handheld Industrial Computer. It contains commands to manage program files, enter and change program lines, and test and debug programs. To accomplish these functions, HXBASIC has three operating modes, each with its own set of commands. The figure and text below show how these modes interrelate. Each mode has a section in this chapter devoted to describing in detail the commands available in that mode.



HXBASIC Command Structure

A user enters the file management mode upon invoking HXBASIC. Its prompt is a dollar sign (\$). Capabilities include creating and deleting files, changing file names, and generating file listings and cross-reference tables. Its commands also invoke the other two modes whose commands are only one character long.

The EDIT command invokes the file-edit mode whose prompt is the percent sign (%). Edit mode commands create new programs via keyboard entry or by merging existing programs. They also allow changes to existing programs. Program syntax is checked with each program line entry or change.

The program debugging mode with its prompt, the at sign (@), is invoked through the

when the debug option /D is used. Various commands are available to set and release breakpoints, specify tracing, and display contents of variables.

Assembly language routines can be called from BASIC language programs. Development and testing of those routines, however, takes place outside the HXBASIC programming environment, using text editors and Intel 8088 assemblers such as the Vectra MS-DOS Macro Assembler (HP 45953A). Refer to the HP-94 Technical Reference Manual for details.

When a syntax error occurs, HXBASIC beeps and places the cursor under the error position. HXBASIC also beeps when you type past the end of a line.

Starting HXBASIC

The BASIC program development utility is the program HXBASIC, contained on your Software Development System disc. HXBASIC will run under MS-DOS Version 2.0 and later on the Hewlett-Packard Vectra and the IBM PC, PC/XT, and PC/AT computers.

Before starting HSBASIC, be sure your CONFIG.SYS file contains the following:

FILES = 18

NOTE 18 is the number of files you can have open simultaneously.

Any informational or error messages you may see are self-explanatory.

HXBASIC does not support HP-IL printers.

The HXBASIC command invokes the BASIC program development utility. It optionally begins execution of a specified BASIC program and a specified run time library.



Examples

```
HXBASIC
HXBASIC filename
HXBASIC B:filename
HXBASIC /library \Direct1\Direct2\filename
```

Description

The optional libraries are files which contain sets of provided or user-written assembly language subroutines. Each library must be located in the BIN subdirectory of the root directory and have the extension LIB. The subroutines are invoked from BASIC programs through the &CALL statement. If a program name is specified, it is loaded and begins to execute upon BASIC start-up. If the file name is used alone (rather than with an MS-DOS path), the HXBASIC operation uses the current working directory. All BASIC program files *must* have the extension BAS.

If no program is specified, HXBASIC enters the file management mode and displays the dollar sign (\$) prompt.

...HXBASIC

Related Commands

BYE

File Management Commands

The file management commands deal with files in their entirety. This mode is the one through which HXBASIC is started and ended. It can be identified by its dollar sign prompt (\$). These eleven commands are available:

BYE	Leave HXBASIC and return to MS-DOS.
DEL	Delete a file.
DIR	Display a directory.
EDIT	Invoke file editing mode.
LIST	List the current program.
LOAD	Load a program from disc to memory.
NEW	Restart HXBASIC.
REN	Change a file name.
RUN	Execute a program and optionally invoke debugging mode.
SAVE	Store the current program to disc.
XREF	Create a cross-reference table.

The BYE command terminates HXBASIC and returns to MS-DOS command mode. It erases the program in computer memory and clears the screeen.



Examples

BYE

Description

The display is cleared and the MS-DOS prompt appears in the upper left hand corner of the screen. Executing BYE erases the current program in memory and all variable assignments made within programs.

CAUTION If the current program in memory has not been saved to disc, it will be lost.

Related Commands

E, HXBASIC

The DEL statement deletes the specified file from disc.



item	Description	Range
path	literal; legal MS-DOS path	
filename	literal; legal MS-DOS filename	8 characters maximum; wild card specifications are allowed
extension	valid MS-DOS file name extension	3 characters maximum; wild card specifications are allowed

Examples

```
DEL filename
DEL B:\direct1\direct2\filename.ext
```

DEL TEST?.B*

Description

If the file name is used alone rather than with an MS-DOS path, the file must be located in the current working directory.

A deleted file can no longer be accessed. The space previously occupied by the file becomes available for creation of other files. Use of a wild card specification allows you to delete more than one file with a single DEL command. An asterisk (*) refers to any string of characters and a question mark (?) represents any single character. For example, *. BAS refers to all files with the extension BAS. TEST?. B* would refer to all files whose name is TEST followed by any single character and whose extension starts with the letter B.

HXBASIC Program Development Utility

...DEL

Related Commands

REN

.

DIR F9 path filename extension Description item Range literal; legal MS-DOS path _ _ path filename literal; legal MS-DOS 8 characters maximum; wild card specifications filename are allowed extension 3 characters maximum; literal: legal MS-DOS file extension wild card specifications are allowed

The DIR statement displays the contents of the specified directory or the directory information associated with a particular file or set of files.

Examples

DIR d: DIR d:dir2 DIR *.BAS

Description

DIR displays the name of the disc drive of the specified directory and a list of the directory contents. The directory entry for each file contains the file name, extension, size in bytes, and date created or last updated. For subdirectories, <DIR> is shown in place of the size.

When DIR is executed without parameters, the contents of the current working directory is listed. If the file name is used alone rather than with an MS-DOS path, the file must be located in the current working directory. Be sure to end the path name with a back slash (\). Use of wild card specification allows you to select which files will be shown in the directory listing. An asterisk (*) refers any string of characters and a question mark (?) represents any single character. For example, *.BAS refers to all files with the extension BAS.

Related Commands

None

Ente EDIT F2 filename path .BAS Item Description Range literal; legal MS-DOS path path filename literal; legal MS-DOS 8 characters maximum filename

The EDIT command starts the program editing mode after optionally retrieving a specified BASIC program file from disc and loading it into system memory.

Examples

EDIT EDIT filename EDIT B:filename EDIT \direct1\direct2\filename

Description

Details about HXBASIC program editing commands are described in the next section of this manual. If no filename is specified, the program currently in memory is used. If the file name is used alone (rather than with an MS-DOS path), the EDIT command uses the current working directory. All (BASIC) program files *must* have the extension BAS, although it is not necessary to specify the extension in the EDIT command.

Specifying a filename will cause EDIT to clear any BASIC programs, subprograms, and variable assignments in memory, and load the specified programs from disc.

CAUTION If the program currently in memory has not been saved to disc, it will be lost even if it is an edited version of the program whose name you specified in the EDIT command.

When the specified program is not on disc, it will create a new program in memory and display the message Creating program <filename>. The system then enters the line input mode and the editor's automatic line numbering function prompts 10 as the first line number. Like any

2-10 HXBASIC Program Development Utility

other program in memory, the program being created will not be written to disc until you explicitly save it using the SAVE command. When the specified program is on disc, the message Editing program <filename> is displayed. The system lists the first twenty lines of the program and displays the % prompt as it awaits program edit mode commands.

This command cannot be used on secured programs.

Related Commands

None

The LIST command lists the current program in memory to the display, a printer, or a disc file.



ltem	Description	Range
device name	literal; legal MS-DOS device name	
beginning line number		0 through 32,767
ending line number		0 through 32,767

Examples

LIST LIST 40,100 LIST ,100 LIST 40, LIST 40 LIST PRN: 100, LIST C:

Description

The range of lines to be listed can be specifed in various ways:

- If both beginning and ending line numbers are specified (as in LIST 40, 100), that portion of the program will be listed.
- Omitting the ending line number (as in LIST 40,) causes listing to start with the beginning line number and to continue through the last line of the program.
- If no beginning line number is specified (as in LIST , 100), the listing begins with the first line of the program and continues through the ending line number.

2-12 HXBASIC Program Development Utility

- When both the ending line number and the comma are omitted (as in LIST 40), only the single line specified is listed.
- Omitting all range parameters (LIST) lists all program lines.

The program name and disc file size (in bytes) precede the list of program lines. Any listing can be aborted by pressing CTRL C or Break CTRL ScrLck.

When you list to a printer, all MS-DOS device names are legal. These include COM1:, COM2:, COM3:, COM4:, LPT1:, LPT2:, LPT3:, PRN:, and AUX: -- use the name configured as your print device.

NOTE Lines longer than the number of characters in a printed line will wrap around to the next line only if the printer does the wrapping. Be sure to enable end-of-line wraparound or compressed print on your printer if you expect your program lines to be longer than the printer's line length. On Hewlett-Packard printers you do this by sending the escape sequence [ESC]&SOC.

To list a program to a disc file, specify the name of the disc (LIST C:, for example). HXBASIC will create a file whose filename is the name of the program and whose extension is LST. If you specify the current drive, the file will be put into the current working directory; for a different drive it will appear in the drive's root directory. The LIST command is a very useful tool for producing a form of a program that can be modified with a text editor. Later, the G program editing command can be used to load the text back in as a program.

This command cannot be used for secured programs.

Related Commands

L, XREF

The LOAD command retrieves the specified BASIC program file from disc and loads it into system memory.



ltem	Description	Range
path	literal; legal MS-DOS path	
filename	literal; legal MS-DOS filename	8 characters maximum

Examples

LOAD filename LOAD B:filename LOAD \direct1\direct2\filename

Description

If the file name is used alone (rather than with an MS-DOS path), the LOAD operation uses the current working directory.

LOAD clears any BASIC programs, subprograms, and variable assignments in memory. All BASIC program files *must* have the extension BAS, although it is not necessary to specify the extension in the LOAD command.

Related Commands

G, M, SAVE

2-14 HXBASIC Program Development Utility

The NEW command restarts HXBASIC. It erases the program in computer memory and clears the screeen.



Examples

NEW

Description

Executing NEW erases the current BASIC program in memory and all variable assignments made within programs. Libraries loaded when HXBASIC was originally started remain loaded.

CAUTION If the current BASIC program in memory has not been saved to disc, it will be lost.

Related Commands

BYE

The REN statement renames a specified file on disc.



item	Description	Range
old path	literal; legal MS-DOS path	
old filename	literal; legal MS-DOS filename	8 characters maximum
extension	literal; legal MS-DOS extension	3 characters maximum
new path	literal; legal MS-DOS path	
new filename	literal; legal MS-DOS filename	8 characters maximum

Examples

REN oldname newname REN d:oldname newname

Description

REN removes the old name from the directory and replaces it with the new name. No wild card specifications are allowed. If the old file name is used alone rather than with an MS-DOS path, the file must be located in the current working directory.

NOTE REN also moves files to a different directory if the old and new paths are not identical. This is different from the MS-DOS command RENAME.

Related Commands

None

2-16 HXBASIC Program Development Utility

The RUN command starts program execution from the beginning, after having loaded the program file from disc if necessary. The debugger is optionally enabled.



ltem	Description	Range
path	literal; legal MS-DOS path	
filename	literal; legal MS-DOS filename	8 characters maximum
/D	literal; invokes debugger	

Examples

RUN RUN <filename> filename RUN /D RUN filename/D /D

Description

Execution of RUN occurs in two steps--variable initialization and program execution. During variable initialization, memory is allocated to all program variables. Variables are set to 0 and the null string.

If an error is detected, initialization halts and an error message is returned. When variable initialization is completed, program execution begins.

Specifying a file name will load that program from disc after clearing the program currently in memory.

CAUTION If the current program in memory has not been saved to disc, it will be lost.

If no file name is specified, the program currently in memory is executed. The /D option starts the debugger. The section "Debugging Commands" describes in detail the commands available in this mode.

NOTE The keyword RUN is optional. Entering only the file name implies execution of the program. Simply pressing Enter will start the program currently in memory.

Pressing CTRL C or Break CTRL Lck will halt running programs. To stop a program that is executing a recursive user-defined function or that has closed the console (CLOSE #0), you must press CTRL Att

CAUTION The program currently in memory will be lost.

Related Commands

None

The SAVE command stores the BASIC program currently in memory into a disc file of the specified name.



anye
ers maximum

Examples

```
SAVE filename
SAVE B:filename,SECURE
SAVE \direct1\direct2\filename
```

Description

If the file name is used alone (rather than with an MS-DOS path), the SAVE operation uses the current working directory. When SAVE is executed, the system searches the specified directory for a BASIC program file with the indicated name. If the file is found, the current program is stored in that file, overwriting the previous contents. The program in memory is not deleted, but its name is changed to that of the disc file. If no such file is found, the file is created in that directory. If no file name is specified, the name of the program in memory is automatically used, and the program is stored in the current working directory.

The maximum size of an individual BASIC language program or subprogram on the development system is 32K bytes. Multiple sub programs can be combined to develop larger applications. Maximum application size is limited by the amount of memory in the HP-94. When SECURE is specified, the XREF, LIST, and EDIT commands cannot be executed for the program stored on the disc. The message SECURED PROGRAM! is displayed if one of these commands is issued for a secured file. The program in memory is not secured. **CAUTION** Once a program is secured, it cannot be released.

When processing the SAVE command, the following confirmation message is displayed: Save to < filename > (Y/N)? Respond Y Enter to store the program. Respond N Enter to abort the command.

Related Commands

LOAD, LIST

The XREF command creates a cross-reference table of all the variables and user-defined functions in the current program.



ltem	Description	Range
device name	literal; legal MS-DOS device name	

Examples

XREF XREF PRN: XREF C

Description

The XREF table contains the name and disc file size of the program followed by information about each program variable: the name of the variable or user-defined function and line numbers referencing it. Variables are listed in the order in which they appear in the program. Function definitions (DEF FN statements) and function calls (FN) are reported as references. The formal parameters for DEF FN statements are shown in the table with a period (.) before the parameter name.

The device name can be used to direct the output to a disc file or a printer. When a disc device is specified, the command places the cross-reference output into a file whose name is the program name and whose extension is REF. If you specify the current drive, the file will be put into the current working directory; for a different drive it will appear in the drive's root directory. When the device name is omitted, the display is used.

When you list to a printer, all MS-DOS device names are legal. These include COM1:, COM2:, COM3:, COM4:, LPT1:, LPT2:, LPT3:, PRN:, and AUX: -- use the name configured as your print device.

This command cannot be used on secured programs.

Related Commands

...XREF

Related Commands

LIST, L

2-22 HXBASIC Program Development Utility

Program Editing Commands

Program editing commands let you create and modify BASIC language programs. The EDIT command starts the program editing mode and displays its prompt, the percent sign (%). You can then input program lines directly from the keyboard, or use any of the following nine commands.

С	Choose a specific line for modification.
D	Delete a program line.
E	Exit from program editing mode and return to file management mode.
G	Get a text file and enter its contents as BASIC program lines
i	Insert new program lines with automatic numbering.
L	List program lines.
м	Merge a BASIC program into the current program.
N	Name the program or change its name.
R	Renumber the program.

Entering Program Lines

To enter a program, type a line number (range 0 to 32,767) followed by the BASIC statement(s) on that line. Press Enter to complete the line. If you are using the I command or if you are creating a new file, HXBASIC will supply the line number for you. After you type a program line and press [Enter], HXBASIC checks the line for syntax errors. Syntax errors include spelling, incorrect parameters, and improper use of a BASIC keyword. If HXBASIC does not detect a syntax error, it enters the line as part of the program and designates it as the current line. Many editing commands allow you to omit the line number parameter if you are working on the current line. If HXBASIC detects a syntax error, it does not enter the line as part of the program. It sets the cursor to the position in the line where the first error was detected. Correct the line by typing over it a There are a number of keys you can use while editing a line:

Key	Operation
¢ ₩	Moves cursor right one character position. Moves cursor left one character position. Chooses the previous program line for editing. Chooses the following program line for editing. Backspace moves the cursor left one character position, and writes a space there.
tab>	Moves cursor to the end of the line currently being edited.
tab<	Moves cursor to the beginning of the line currently being edited.
ESC ENTER	Aborts any changes to the line currently being edited, and returns to the % prompt.
Del	Deletes the character to the left of the cursor position, and moves the remaining characters left one column
Ins	Switches between replace character and insert character mode. The default mode is replace character where a new character replaces the existing character at the cursor position. Press- ing ins switches to insert character mode, where a new character is inserted at the blink- ing cursor position, moving the existing charac- ters right one column. Pressing ins again switches back to replace character mode.

.
The C command chooses a specified line for updating.



item	Description	Range
line number	integer constant identi- fying a program line	0 through 32,767
F	first line of the program	
L	last line of the program	

Description

When a program line is chosen using this command, its contents are displayed and the editor enters the line input mode. Correct the line by typing the changes over the line and press [Enter] to enter the correction. If syntax errors remain in the line, the cursor will pause under the portion in error. The correction is not complete until the entire line has correct syntax. If you want to abort a change, press [ESC]. This will end line input mode, restore the % prompt, and leave the line unchanged. When correction is complete, the editor returns to the file editing mode (% prompt). The line just edited becomes the current line. Specifying F refers to the first line of a program. L refers to the last line. If a line number is not specified, the command uses the current line.

Related Commands

យ, ពា

The D command deletes program lines from the current program in memory.



item	Description	Range
beginning line number	integer constant identi- fying a program line	0 through 32,767
ending line number	integer constant identi- fying a program lin e	0 through 32,767
F	first line of the program	
L	last line of the program	

Examples

D 30 D 30,90 D F,100 D 2000,L 30

Description

Specifying only the beginning line number deletes that line. Specifying both parameters deletes all lines within that range.

Specifying F refers to the first line of the program. L refers to the last line.

NOTE The command name D is optional when deleting a single line. Specifying only a line number will delete that line.

Related Commands

DEL

2-26 HXBASIC Program Development Utility

The E command exits program editing mode and returns to HXBASIC file management mode.



Description

The HXBASIC file management prompt \$ appears on a new line. The BASIC program currently in memory is not erased. It remains available for further commands. To save your changes to disc, however, you must execute the SAVE command.

Related Commands

BYE

The G command gets the specified text file from mass storage and attempts to enter the contents into memory as program lines.



item	Description	Range
path	literal; legal MS-DOS path	
file name	literal; legal MS-DOS filename	8 characters maximum
extension	literal; legal MS-DOS extension; default is LST	3 characters maximum

Examples

- G filename.ext
- G \direct1\filename.ext

Description

G retrieves ASCII character strings from the specified ASCII text file. If you do not specify an extension, LST is assumed. Each record is read as a separate character string. When a string consists of a valid BASIC program statement preceded by a line number, the string is entered into system memory as a program line. If a string cannot be properly interpreted as a program line due to a syntax error, reading is stopped temporarily and the line is displayed with the cursor positioned at the error. To correct a line with syntax errors, edit it as you would if you had typed it yourself, and press Enter. HXBASIC will continue to read the text file. To remove a line with errors, press ESC and Enter. HXBASIC will then continue to read the rest of the file.

The retrieved lines are read into system memory without erasing the program already there. If an incoming line has the same line number as a line already in memory, the new line overwrites the original line.

2-28 HXBASIC Program Development Utility

The G command in conjunction with LIST is a very useful tool for modifying a program with a text editor such as Executive MemoMaker (HP68330F) instead of the HXBASIC editor. LIST is used to write out the file for the text editor to work on. Later, the G command can be used to load the text back in as a program. The text file should not contain control characters. If the file contains control characters, unpredictable results will occur.

Related Commands

LOAD, M

.

The I command invokes the line-input mode and provides automatic line numbering during program entry.



ltem	Description	Range
beginning line number	integer	0 through 32,767
increment	integer (default=10)	1 through 32,767

Examples

- I 100,2
- I F,2
- ΙL

Description

Executing I displays the specified beginning line number. When the contents for that line have been entered, a new line number, computed by increasing the current line number by the increment, is displayed.

Automatic line numbering is halted by pressing [Enter] in response to a new line number. If the beginning line number is omitted, auto numbering starts one increment after the last line of the program. If a line with the beginning line number already exists, the command will prompt with a line number one increment higher. The command will end insert mode and return to the edit mode prompt % when it computes a new line number that is equal to or greater than the next line in the program.

Related Commands

None

2-30 HXBASIC Program Development Utility



The L command lists the current program in system memory on the display.

ltem	Description	Range
beginning line number	integer	0 through 32,767
ending line number	integer	0 through 32,767
F	first line of the program	
L	last line of the program	

Examples

L L 40,80

Description

The beginning and ending line numbers specify the portion of the program to be listed. If no ending line number is specified, twenty lines are listed.

When both parameters are omitted, the listing begins at the current line of the program and continues for ten lines. Specifying F refers to the first line of the program. L refers to the last line.

Related Commands

LIST

L

...L

Related Commands

LIST

2-32 HXBASIC Program Development Utility

The M command merges the specified portion of a program retrieved from mass storage with the current program in system memory.



Item	Description	Range
filename	literal; name of a BASIC program file	8 characters maximum
path beginning line number	literal; legal MS-DOS path integer constant identifying a pro- gram line (default=first line number of specified program)	 0 through 32,767
ending line number	integer constant identifying a pro- gram line (default=last line number of specified program)	0 through 32,767

file name	T{ literal; name of a BASIC program file
T}	T{ 8 charact ers maximum T}

Examples

M \direct1\filename
M filename 200,5000

M retrieves the specified portion of a BASIC program file from mass storage and adds it to the current program in system memory. If the optional parameters are omitted, the entire program is merged into the program in memory.

CAUTION If a line in the program being merged from disc has the same number as a line in the program currently in memory, the line from disc will overwrite the one in memory.

When programs are merged using the optional line number parameters, only lines within their range are retrieved from disc. If only the starting line is specified, only that line is retrieved.

Secured programs cannot be merged.

Related Keywords

G, LOAD

The N command displays or changes the name of a program.



ltem	Description	Range
program name	literal; legal MS-DOS filename	8 characters maximum

Examples

N N newname

Description

When this command is executed with a program name, the former program name is displayed in the format Former program name: <filename> and is then changed to the specified name. If the program name is omitted, the file name is displayed in the format Current program name: <filename>, but is not changed.

Related Commands

None



The R command renumbers the current program in memory.

Examples

R R 10,1

R ,100

Description

The entire program will be renumbered. The first line in the program is assigned the new initial line number. Successive lines are renumbered according to the specified new increment value. The line numbers are not changed if renumbering causes the new ending line number to exceed 32,767. When R changes a line number, all programmatic references to that line number within the program (for example, GOTO *line number*) are automatically updated except for references from SYLB and SYSW.

After the command has completed, the first ten renumbered lines are displayed. The last line displayed becomes the current line.

CAUTION Line numbers referenced by SYLB and SYSW will not be changed.

If there are line number references to non-existent lines, these references will not be updated. This may result in a change in program flow.

2-36 HXBASIC Program Development Utility

R

Related Commands

XREF

...R

Program Debugging Commands

Program debugging commands help you test programs to detect coding errors. Using the RUN command with the /D option will invoke the debugger and make various commands available for checking program execution. Program errors are corrected by entering program edit mode and using the commands described in the previous section. HXBASIC's prompt to indicate it is awaiting a debugging command is the at sign (@). When a program begins execution in debugging mode, program execution is suspended and the debugging prompt @ is displayed together with the program name, line number, and statement number (for example, PROGRAM.1 0.2@). Statement numbers (starting with 0) are assigned sequentially to statements on a line. Debugging facilities available include breakpoints, statement execution tracing, and display of program variables. The specific debugging commands available are shown in the table below.

В	Set a breakpoint.
D	Display the contents of a variable.
E	End program debugging mode and return to file management mode.
R	Resume program execution.
Т	Specify section of the program to be traced.
U U	Clear a breakpoint.
[Enter]	Execute the next statement.

The B command sets a breakpoint at the line specified.



Examples

B 40 B 40.2 B program.50

Description

The B command specifies where breakpoints will be set to halt program execution. Actual breakpoint execution begins when program execution is restarted. Immediately before executing a statement set as a breakpoint, the system enters the debugging command wait state (signified by the at sign prompt **@**). Once the program has stopped at a breakpoint, it can be restarted with the R resume command or one statement at a time can be executed by pressing [Enter] (the single-step command). Only one breakpoint can be set with each B command, but use of several B commands can activate up to four breakpoints in a program. B commands that exceed the breakpoint limit cause an error and are ignored. The program name option is useful for programs calling a number of subprograms. The option specifies the program or subprogram to which the breakpoint applies. If the program name is not specified, it defaults to the program whose execution is currently suspended.

The line number specifies the line in the program where the breakpoint will be set. The optional statement number further specifies where on the line the breakpoint will occur. If a statement number is not specified, it defaults to the first statement on the line (statement zero). Line and statement numbers are not checked to see if the line or statement actually exist in the program.

Using the B command without a line number will display the currently set breakpoints, separated by colons. Breakpoints are cleared by executing the U command. All breakpoints are automatically cleared when the program ends.

Related Commands

R, U

The D command displays the current value of a variable.



ltem	Description	Range
variable name	literal; name of a BASIC variable	any valid name
subscript	integer	0 through 32,767

Examples

D X D X(5)

Description

The variable name is the name of the variable whose value will be displayed. Be sure to include the subscript for array variables. The display format for numeric variables is the same as that of the BASIC language statement PRINT without a format. For string variables, the hexadecimal value of the string is displayed followed by the ASCII interpretation. The string "xyz" would be displayed as:

Hex: 78797A ASCII: xyz

If a specified variable is not defined or not used in a program, an error message appears.

Related Commands

None

D

The E command exits program debugging mode and returns to HXBASIC file management mode.



Description

The HXBASIC file management prompt \$ appears on a new line. The BASIC program currently in memory is not erased. It remains available for further commands.

Related Commands

BYE, RUN

The R command resumes program execution.



Examples

R

Description

The R command resumes program execution starting at the statement whose line number and statement number are being displayed. When a statement has been set as a breakpoint, the program will stop just prior to that statement's execution and await a debugging command (at sign @ prompt). When executing program sections specified for tracing with the T debugging command, the line and statement numbers are displayed as each statement of the trace section is executed.

Related Commands

B, T, [CTRL] [A], [CTR] [C]

Т

The T command specifies a section of a program to be traced.



Item	Description	Range
program name	literal; name of a BASIC program or subprogram	any valid name
starting line number	integer constant	0 through 32,767
ending line number	integer constant	0 through 32,767
negative line number	integer constant	0 through 32,767

Examples

T PROGRAM.50, 100

Description

The T command merely specifies the section of the program to be traced. Actual trace execution begins when program execution is restarted using the R command. Tracing will not occur while single-stepping a program. During trace execution, the line and statement numbers of each statement are displayed as they are executed.

The program name option lets you specify tracing within a subprogram. If the program name is not specified, it defaults to the program whose execution is currently suspended.

Only one trace section can be specified. If more than one trace section is specified, only the last one is used.

The starting and ending line numbers specify the lines in a program whose execution will be traced. Tracing is limited to the lines in the program specified. If the line range includes a CALL to a subroutine, that subroutine's execution will not automatically be traced. When only the starting line is

2-44 HXBASIC Program Development Utility

specified, only execution of that line will be traced. The starting line number must be less than the ending line number. If the line range specified does not exist in the program, no tracing will be performed for it.

Using the T command without a line range will display the currently set tracing specification. Tracing operations are cancelled by executing the T command with any negative line number. It is not necessary to specify the program name when cancelling trace operations with a negative line number.

Related Commands

R





ltem	Description	Range
program name	literal; name of a BASIC program or subprogram	any valid name
line number statement number	integer integer	0 through 32,767 0 through 32,767

Examples

U U 40 U 40.2 U program.50

Description

The U command clears a breakpoint set earlier with the B comma nd. Only one breakpoint can be specified with each U command, but use of several U commands can clear multiple breakpoints in a program. If the line and statement numbers are omitted, all of the breakpoints are cleared. The program name is useful for programs calling a number of subprograms. The option specifies to which program or subprogram the breakpoint applies. If the program name is not specified, it defaults to the program whose execution is currently suspended. The line number and statement number identify the specific breakpoint to be cleared. If the specified breakpoint does not exist, an error message is displayed followed by a list of currently set breakpoints, separated by colons. All breakpoints are cleared automatically when the program ends.

Related Commands

B, R

U

The single-step command executes the current statement whose number is being displayed and then enters debugging command wait state.



Description

The singlestep command has no keyword. Simply press [Enter] to execute the next line of the program.

Related Commands

R

[CTRL] [A]

The [CTRL][A] command interrupts the program for debugging.

Examples

[CTRL] [A]

Description

The [CTRL] [A] command interrupts the program currently executing and invokes the debugging mode. If the program is waiting for input or output, the input or output must be completed before the program will stop.

Related Commands

[CTRL] [C], [CTRL] [S], B, R

The [CTRL][C] command aborts program execution.

Examples

[CTRL] [C]

Description

Use of the [CTRL] [C] command aborts program execution. Like [CTRL] [Scr Lck], this operation may also be used to end the output of lists generated by the LIST and L commands.

Related Commands

[CTRL] [A]

HXBASIC Program Development Utility 2-49

[CTRL] [P]

The CTRL P command produces a printed copy of the information as it appears in the display.

Examples

CTRL P

Description

The CTRL P command directs a copy of all currently displayed information to the currently configured development system printer.

NOTE This is different from the MS-DOS command CTRL P.

Related Commands

LIST, XREF

2-50 HXBASIC Program Development Utility

The [CTRL][Q] command restarts output to the alpha display.

Examples

[CTRL] [Q]

Description

Use of the [CTRL] [Q] command resumes output to the display after it has been suspended by the [CTRL] [S] command.

Related Commands

[CTRL] [S]

[CTRL] [S]

The [CTRL][S] command temporarily suspends output to the display.

Examples

[CTRL] [S]

Description

Use of the [CTRL] [S] command temporarily "freezes" the screen. The [CTRL] [Q] command will release the screen to continue output.

.

Related Commands

[CTRL] [Q], [CTRL] [A], [CTRL] [C]

HXC File Conversion Utility

-

Contents

Chapter 3

HXC File Conversion Utility

- 3-1 Introduction
- **3-2** HXC
- 3-2 Examples
- 3-2 Description
- 3-4 Running HXC Interactively
- **3-4** Moving the Display and Cursor
- **3-4** Stopping HXC
- **3-5** Specifying the Input Files
- **3-6** File Limits
- **3-7** Specifying Parameters for RAM Output
- **3-9** Specifying Parameters for ROM Output
- **3-11** Running HXC Automaticlly
- 3-12 How to Write Command Files
- **3-12** Writing Command Files for RAM Output
- **3-13** Writing Command Files for EPROM Output
- 3-15 About Error Messages
- **3-15** About Making EPROMs
- 3-15 Output File Formats
- **3-15** Map File Format (MMP)
- **3-15** Map File Format for BASIC Programs (BMP)
- **3-17** MDS File Format for RAM Output
- **3-20** MDS File Format for ROM/EPROM Output
- 3-21 Backup Files

HXC File Conversion Utility

Introduction

Application programs for the HP-94 are created and assembled on a development computer using such tools as HXBASIC or the Vectra MS-DOS Macro Assembler (HP 45953A). These programs, as well as any data files, have to be converted into a form the HP-94 can use before being transmitted to the HP-94's RAM or to an EPROM programmer.

HXC performs the conversion and combines the files specified into a single file in Intel MDS format, an error-detecting (but not error-correcting) format. HXCOPY (see chapter 4) completes the transmission by performing the actual transfer to the destination device.

HXC performs four tasks. It:

- Converts BASIC programs into forms which can be used by the HP-94 BASIC interpreter.
- Converts input files into the MDS format required by the HP-94 for transmission error detection.
- Combines two or more component files into a single output file for convenient, one-step transfer of the component files to the HP-94.
- Creates ROM image files for loading into ROM or EPROM. (The integrated circuit is then inserted into an HP 82412A ROM/EPROM Card which can be plugged into an HP-94D or HP-94E.)

You can use HXC in two different ways:

Interactively

When you use this mode, you provide HXC information about the component files by entering the information into the development-system computer.

Automatically

When you use this mode, you provide HXC with a command file describing all of the component files. You can create a command file with a text editor, or you can run HXC interactively, and HXC will create it for you.

HXC

HXC is a utility program contained on your Software Development System disc. It is supported on the HP Vectra and IBM PC, PC/XT, and PC/AT computers running the MS-DOS operating system version 2.0 or later.

Be sure your CONFIG.SYS file contains the following:

FILES=9 DEVICE=ANSI.SYS

NOTE Nine is the number of files you can have open simultaneously. If you are running other utilities, you may wish to set this figure at a higher value, such as 18 for HXBASIC. Then you can run the other utilities as well as HXC.



ltem	Description	Range		
MDS file	literal; legal MS-DOS path name (no exten- sion required)	maximum 20 characters		
command file	literal; legal MS-DOS path name (no exten- sion required)	maximum 20 characters		

Examples

HXC XFER HXC @XFER

Description

HXC will process files for transfer either to the memory in an HP-94 or to an EPROM programmer. The MDS file may also be sent on a disc or in an EPROM to an integrated-circuit manufacturer for incorporation into a ROM.

The name of the MDS file specified in the command to run HXC is used to name the output file created by HXC.

Specifying an existing MDS file retrieves the information used to create that file.

3-2 HXC File Conversion Utility

When HXC is run interactively, it reads information about the component files from the CMD file. If no CMD file is present, HXC will read as much information as it can from the MDS file. This information will only describe input files that converted successfully, since the MDS file does not contain any files that failed to convert.

A command file is generated automatically when HXC runs. It is a text file with the MDS file name and the extension CMD. It contains the names of all of the files you input interactively plus additional information about the files. The next time you create a new version of the output file, HXC will look in the command file rather than ask you to input the information again.

Input files fall into three categories:

- BASIC intermediate language files created by HXBASIC. They always have the extension BAS and are converted to a special file form for use by the HP-94 BASIC interpreter.
- Assembly language programs which have been assembled and linked. These have the extension EXE.
- Data files. Any files which do not have the extensions BAS or EXE are considered data files and are converted directly into Intel MDS format.

Each time HXC is run, three output files are created. All share the same file name, but each has a different extension. The file name assigned to these files is the name of the MDS file which you used to start HXC or the new output filename you selected when prompted.

- The MDS (Microprocessor Development System) file (extension MDS) is the file ready to be transmitted to the HP-94 or to the EPROM programmer.
- The MBK (Mds BacKup) file is a backup file (extension MBK). If a file with the name of the MDS file already exists, it is renamed to the extension MBK when HXC ends normally.
- The MMP (Mds MaP) file (extension MMP) is a text file containing the information provided to HXC as well as program size and error information.

In addition, if you run HXC interactively, it creates two more files:

- The CMD (CoMmanD) file (extension CMD).
- The CBK (Command BacKup) file (extension CBK) which contains the information in the MDS file.

If any of the input files are BASIC programs, a BMP (Basic MaP) map file (extension BMP) is created for each BASIC program being converted. It contains program and variable sizes as well as error information.

You can use the MS-DOS commands TYPE or PRINT to examine the files.

NOTE You must specify an MDS file name even if you are not editing an existing MDS file or using a command file.

Running HXC Interactively

HXC uses a screen-oriented user interface. When HXC starts running, it displays a screen in which function-key menus are displayed and certain information concerning the input files is requested. You input the information, and HXC then produces a single output file.

Moving the Display and Cursor

You can use the following keys to move the cursor around the screen, to scroll portions of the screen, and to change the displays.

Key	Function				
[DEL]	Erases the character on which the cursor is located.				
(ESC)	Erases the line in which the cursor is located.				
(†)	Moves up one line, and scrolls a portion of the screen down.				
ш	Moves down one line, and scrolls a portion of the screen up.				
[<-]	Moves left one character.				
[->]	Moves right one character.				
[Pg Dn]	Displays the next screen.				
[Pg Up]	Displays the previous screen.				
[ENTER]	Moves to the left of the next line, and scrolls a portion of the screen up.				

Stopping HXC

By pressing [F1] or [CTRL] [C1, you stop HXC without performing any conversions and return to MS-DOS.

Specifying the Input Files

NOTE Assume you wish to name your output file XFER.

A. Run the HXC program by typing HXC XFER and pressing [Enter]. The display on your screen will look like this:

HXC (A.01.06) HP825 Output file(s) xfer.MDS	20 (c) Copyr	ight Hewle	tt-Packard Status Files:	1985 : Entering (0	dev. sys. names
Development System File Name	Handheld File Name	File Type	Actual Size(byte)	Allocated Size(para)	Size Increment(para)



The cursor blinks at the top left side of the **Development System File Name** field waiting for you to type in the names of the files you wish to convert.

The name XFER.MDS appears in the Output file(s) field.

Entering dev.sys. names appears in the Status field.

B. Type in the name of your first input file.

HXC will accept three types of files. It identifies them by their extensions as listed below and certain information at the beginning of each file.

 BASIC language intermediate files (extension BAS) which were created by the HXBASIC Program Development Utility (see chapter 2).

- Assembly-language programs (extension EXE) which were assembled and linked so as to be in executable form.
- Data files whose extension is anything except BAS or EXE (including no extension).

When you get to the end of the name, press either [Enter] or [Tab]; the cursor will proceed to the beginning of the next line and wait for you to type in your next path name.

File Limits

The maximum number of RAM files you may have is:

- 63 for the HP-94D.
- 63 for the HP-94E.
- 127 for the HP-94F.

The maximum number of ROM files you may have is:

- 31 for a 32 or 64 K-byte ROM or EPROM.
- 63 for a 96 or 128 K-byte ROM or EPROM.

If you enter more than 127 RAM files or more than 31 or 63 ROM files, an error message is displayed.

Things To Remember

- Enter the full path name of each file, including subdirectories if necessary. (If a file name with no path is used, the current directory is searched for the file.)
- The full path name may not exceed 20 characters.
- Each path name must be on a separate line.
- Only ten path names can be displayed at a time. (When you press [Enter] to enter your eleventh and succeeding path names, the list of files will automatically scroll up to make room for the new names).
- To scroll the list of more than ten file names down, run the cursor to the top of the list; to scroll the list up, run the cursor to the bottom of the list. Pressing (Pg Up) and (Pg Dn) will move the cursor to the top and bottom of the list, ten lines at a time.
- You can erase any path name you have entered by moving the cursor to the line you wish to remove and pressing [CTRL] [U].
- If you transfer a file from the HP-94 to your development-system computer, it will have the handheld file name and type. If you then convert this file, HXC will use this information to generate an appropriate development-system file name.
- Only data files are allowed to be zero-length files.
- **C**. Repeat step B for the rest of your file names.

The task of specifying the input files is now complete.

If you wish to transfer your output file into RAM in an HP-94, go to Specifying Parameters for RAM

3-6 HXC File Conversion Utility
Output.

If you wish to transfer your output file to a ROM or to an EPROM programmer, go to Specifying Parameters for ROM Output.

Specifying Parameters for RAM Output

The cursor is flashing at the bottom of the list of input files.

A. Press [F2]. "RAM MDS file" appears in the MDS summary field in the top center of the screen.

If you have not used the name of an existing MDS file for the name of your output file, go to step C. If you have used the name of an existing MDS file, the cursor proceeds to top-center screen, and you are prompted for an output file name.

B. Do one of the following:

Type a new output file name, and press [Enter]. (The new name replaces the old name in the Output file(s) field).

or

Press [Enter] if you wish to use the existing file name.

The cursor proceeds to the top of the Handheld File Name field. The table below defines the information you will be prompted to input.

Item	Description	Range
Handheld File Name	literal; file name on HP- 94 (default is the first 4 characters of the development system file name)	4 alphanumeric charac- ters maximum; first character alphabetic
File Type	literal; file type on HP- 94	1 character; A, B, D, or H
Allocated Size(para)	data file initial size in handheld; in para- graphs hexadecimal (default is the actual size rounded up to the nearest paragraph)	0h-FFFFh, greater than or equal to actual size
Size Increment(para)	increment by which the data file will expand when data is entered into it; in paragraphs hexadecimal (default is 1)	0h-FFFFh

NOTE A paragraph is 16 bytes long.

C. Type in the name you wish to use corresponding to your first development-system file and press [Enter]. (If you do not type in a name before pressing [Enter], the first four characters of the development-system file name will be used.)

- If your file is a BASIC file (extension BAS), the name is entered, a B is entered in the File Type field, and the cursor returns for the next file name.
- If your file is a data file (no extension or an extension other than BAS or EXE), the name is entered, a D is entered in the File Type field, the actual size of the file is entered in the Actual Size(byte) field, and you are prompted for an input in the Allocated Size(para) field.

Enter the allocated size, and press [Enter]. You are prompted for an input in the Size Increment field.

Enter a size increment, and press [Enter]. The cursor returns for the next handheld file name.

• If your file is an assembly-language file (extension EXE), the name is entered, and you are prompted in the File Type field to enter A, D, or H.

Enter A if your file contains an application program or a BASIC keyword.

Enter D if your file is a data file.

3-8 HXC File Conversion Utility

Enter H if your file contains a device handler.

After entering A or H, press [Enter]; the cursor returns for the next handheld file name. D is processed as described above.

Note that any information entered in lowercase characters is automatically converted to uppercase characters.

D. After you have entered all the required information, the conversion process begins automatically.

At the end of the conversion:

- The numbers of files converted and not converted are displayed in the Status field.
- If a file failed to convert, an error message tells you why it failed.
- The conversion results are repeated at the bottom of the screen, and
- The space required in the handheld is displayed.

Specifying Parameters for ROM Output.

The cursor is flashing at the bottom of the list of input files.

A. Press [F3]. You are prompted in the upper center screen for a ROM size in K bytes.

NOTE The ROM size may be 32, 64, 96, or 128 K bytes.

B. Enter the ROM size, and press [Enter].

If you entered 128, ROM MDS file and Directory 1 (128 KB) appear in the MDS summary field.

You are prompted in the upper center screen for four different output file names (one of the four may be the original output file name).

C. Do one of the following:

Type a new output file name, and press [Enter].

or

Press [Enter] if you wish to use the original file name.

NOTE If the ROM contains 128 K bytes, you will be prompted for three additional output file names.

If the ROM contains less than 128 K bytes, you will be prompted for additional output file names and the directory numbers desired. This information is summarized in the table below. Refer to the *HP-94 Technical Reference Manual* for a further understanding of directory numbers and how to use them.

ROM Size in K Bytes	Possible Directory Numbers	Number of 32 K-Byte Output Files
32	1-4	1
64	1-3	2
96	1-2	3
128	1	4

The new name(s) replace the original name in the Output file(s) field.

After you have entered the required information, press [Enter]. The cursor proceeds to the Handheld File Name field.

item	Description	Range
Handheld File Name	literal; file name on HP- 94 (default is the first four caracters of the development-system file name)	4 alphanumeric charac- ters maximum; first character alphabetic
File Type	literal; file type on HP- 94	1 character; A, B, D, or H

D. Type in the name you wish to use corresponding to your first development-system file and press (Enter]. (If you do not type in a name before pressing (Enter], the first four characters of the development-system file name will be used.)

- If your file is a BASIC file (extension BAS) or a data file (no extension or an extension other than BAS or EXE, the name, a B for a BASIC file, or a D for a data file, is entered in the File Type field, and the cursor returns for the next file name.
- If your development system file is an assembly-language file (extension EXE), the name is entered, and you are prompted in the File Type field to enter A, D, or H.

Enter A if your file contains an application program or a BASIC keyword.

Enter D if your file is a data file.

3-10 HXC File Conversion Utility

Enter H if your file contains a device handler.

After entering A, D, or H, press [Enter]. The cursor returns for the next handheld file name.

E. After you have entered all the required information, actual and allocated sizes are entered automatically, any size increments are set to 0, and the conversion process starts automatically.

HXC will only use as many output files as are needed to hold the input files. If you have specified more output-file names than are required for the transfer, the extra file names will automatically be deleted from the list in the Output file(s) field, and those files will not be created.

At the end of the conversion:

- The numbers of files converted and not converted are displayed in the Status field.
- If a file failed to convert, an error message tells you why it failed.
- The conversion results are repeated at the bottom of the screen, and
- The space required in the handheld is displayed. (This is the total space required in the ROM or EPROM.

Running HXC Automatically

You have converted a number of files into a single MDS output file. Now you make a change in one of the files (without changing the file name). You need to run HXC again. Instead of running HXC interactively, when you would have to enter all the file information again, you can run HXC using the command-file name, eg. HXC XFER. (The extension is not required.) This causes the new MDS file to be generated automatically using the file names and information in the command file.

To do this, you must first do one of the following:

- Run HXC at least once interactively so that a command file is automatically generated.
- Create a command file using a text editor (refer to the section on how to write command files for details.)

How to Write Command Files

A command file for a given set of input files can be created by running HXC interactively. However, you may wish to edit your command files sometime, so the following information on writing command files is provided.

You may use any text editor so long as the resulting file is in standard ASCII format.

Writing Command Files for RAM Output

The sample file below illustrates a command file. The comments printed in smaller type are for illustration purposes only. Do not include comments in your file.

```
RAM ← Specifies RAM MDS file
Sample ←Output file name
★ ←Indicates start of input file information
a.exe(main)
                      A \leftarrow File type
b.exe(sub1)
                      Η
                              Allocated size
c.exe(sub2)
                      а
                              T
d.exe(sub3)
                      d
                              10
                                     1 \leftarrow Size increment
e.exe(sub4)
                      d
                              0
                                     а
                      d
abc0(dat1)
                              100
                                     d
abc1(dat2)
                      d
                                     2
                             b0
abc2(dat3)
                              f0
                                     10
                      d
abc3.bas(sub5) B
abc4.bas(sub6) B
abc5.dat(sub7) d
                                     5
                              1
*
              <sup>†</sup>Handheld file name
TIndicates end of input file information
```

Things to Remember

- The first line calls for the creation of a RAM file.
- The second line contains the name of the output MDS file to create.
- The third line is an asterisk to indicate the beginning of the component-file block.
- Each line in the component file block provides the same information called for when using HXC interactively. The format of the information, however, is slightly different.
- Specify only items 1 through 3 below for programs (EXE or BAS files). All five items are required for data files. They are separated by one or more spaces or tabs.
 - 1. Development System File Name. Include subdirectory path, if required, but do not exceed 20 characters.
 - 2. Handheld File Name. This name immediately follows the Development System File Name (no spaces or tabs) and is contained in parentheses. File names on the HP-94 are limited to four alphanumeric characters; the first character is alphabetic. The name can be entered in lowercase.
 - 3. File Type. Valid options are:

A for assembly language EXE files containing application programs or BASIC keywords.

3-12 HXC File Conversion Utility

B for BASIC language BAS files.

D for data files.

H for assembly language EXE files containing device handlers.

- 4. Allocated Size. Specify for data files only. It is the initial size for the file on the handheld. Enter the size in 16-byte paragraphs hexadecimal. The size cannot be smaller than the current file size.
- 5. Size Increment. Specify for data files only. It is the size by which the file on the HP-94 will increase as the file grows. Enter the increment in 16-byte paragraphs hexadecimal.
- Another asterisk ends the component-file block.

Writing Command Files for EPROM Output

Command files specifying output for EPROM programmers are very similar to the command files for RAM output described in the previous section.

ROM/EPROM command files differ from those for RAM in three areas:

- Line one specifies ROM instead of RAM.
- Line one also specifies the ROM size and directory number.
- Due to the read-only nature of ROM and EPROM devices, no allocated size or size increment is specified.

The sample file below illustrates a command file for ROM/EPROM output. The comments printed in smaller type are for illustration purposes only. Do not include comments in your file.

```
ROM 32
            4 ← Specifies ROM MDS file, ROM size in K bytes, and directory number
Sample \leftarrow MDS file name (there may be 3 more depending on the ROM size)
★ ←Indicates start of input file information
a.exe(main)
                     A ← File type
b.exe(sub1)
                      Н
c.exe(sub2)
                      а
d.exe(sub3)
                      d
e.exe(sub4)
                      d
abc0(dat1)
                      d
abc1(dat2)
                      d
abc2(dat3)
                      d
abc3.bas(sub5) B
abc4.bas(sub6) B
abc5.dat(sub7) d
*
              <sup>†</sup>Handheld file name
```

\uparrow Indicates end of input file information

Things to Remember

The first line calls for the creation of a ROM image, specifies its size, and lists the number of the HP-94 directory in which the resulting MDS file is to be placed. Refer to the discussion of memory management in the HP-94 Technical Reference Manual for help in selecting the proper

values. Valid options are:

ROM Size in K Bytes	Possible Directory Numbers	Number of 32 K-Byte Output Files
32	1-4	1
64	1-3	2
96	1-2	3
128	1	4

- The 1 to 4 lines contain the names of the existing MDS file to create.
- The next line is an asterisk to indicate the beginning of the component file block.
- Each line in the component-file block provides the same information called for when you run HXC interactively. The format of the information, however, is slightly different.
- Specify the three options in the list below. They are separated by one or more spaces or tabs.
 - 1. Development System File Name. Include subdirectory path, if required, but do not exceed 20 characters.
 - 2. Handheld File Name. This name immediately follows the Development System File Name (no spaces or tabs) and is contained in parentheses. File names on the HP-94 are limited to four alphanumeric characters; the first character is alphabetic. The name can be entered in lower case.
 - 3. File Type. Valid options are:

A for assembly language EXE files containing application programs or BASIC keywords.

B for BASIC language BAS files.

D for data files.

H for assembly language EXE files containing device handlers.

Another asterisk ends the component file block.

NOTE If you enter information incorrectly into the command file and run HXC interactively, HXC will display a warning that something is wrong.

If you know what is wrong, continue to run HXC interactively, entering the information correctly; HXC will correct the command file.

If you do not know what is wrong, exit HXC, then rerun HXC using the - command name. An error message describing the problem will be displayed.

3-14 HXC File Conversion Utility

About Error Messages

All the error messages you may see are self-explanatory.

About Making EPROMs

Your EPROM programmer must support the Intel MDS format, and must have the ability to program 256 K-bit CMOS EPROMs, such as the 27C256. This will permit you to create up to 96 K-bytes worth of EPROM-based software.

If you wish to create 1228 K-bytes worth of software, your programmer must have the ability to program 512 K-bit EPROMs, such as the 27C512.

The EPROMs must have speed ratings of 2250 ns or less.

When programming a 512 K-bit EPROM, you must place two 32 K-byte MDS output files created by HXC in the programmer together, the first 32 K-byte output file in the first 32K bytes of the programmer RAM, and the second 32 K-byte file in the second 32K bytes of RAM, or you can program the two halves of the EPROM separately. Refer to the operating instructions for you EPROM programmer for further information.

Output File Formats

HXC creates a number of output files. The following are described in this section:

- The map file (extension MMP) (MdsMaP).
- The map file for BASIC programs (extension BMP) (BasicMaP).
- The MDS file for RAM output (extension MDS).
- The MDS file for ROM/EPROM output (extension MDS).
- Backup files for MDS files (extension MBK) (MdsBacKup) and CMD files (extension CBK) (CommandBacKup).

Map File Format (MMP)

HXC creates an MMP file every time it runs. The file is in standard ASCII format, so MS-DOS commands such as TYPE and PRINT may be used to display it. With the exception of the location of several items in the display, the format of the file and information presented are identical to that of the HXC screen.

Map File Format for BASIC Programs (BMP)

HXC creates a special map file for each BASIC program it converts. The filename of each file is that of the name of the BASIC program it describes. The extension is always BMP. The file is in standard ASCII format, so MS-DOS commands such asTYPE and PRINT may be used to display it. The BMP file is divided into two parts:

- Program information.
- A description of each of the variables used in the program.

Program Information

The size in bytes and offset address (when applicable) of the following items of program information are displayed:

- Program
- Executable Code
- Variable Descriptor Table
- Variable Space Required

The size and address are in hexadecimal.

Variable Information

Variables are listed in alphabetical order. The following information about each of the variables used in the BASIC program is displayed:

- Array Elements
- Length
- Starting Address
- Variable Name

Each of the fields is defined as follows:

Field	Description
Array Elements	for arrays, the number of ele- ments in decimal. Field is blank for non-array variables.
Length	the number of bytes required for the variable in decimal. For arrays, this is the total size of the array.
Starting Address	the starting address of the vari- able in hexadecimal. The first four digits are the segment. The last digit is the offset. This seg- ment and offset are relative to the start of the variable area indi- cated by the A command of the HP-94 BASIC debugger, SYBD.
Variable Name	name of the variable

Error Information

If BASIC syntax errors are detected during the HXC conversion, they are listed by themselves in the BMP file.

All error messages you may encounter are self-explanatory, and indicate the line number in which the error was detected.

MDS File Format for RAM Output

The MDS file is the primary output file. It is a single file that contains all of the component files specified for the HXC operation. The MDS file is in a format that can be accepted by the HP-94. Refer to chapter 4, *HXCOPY File Copy Utility*, for details on performing the actual transfer.

The MDS format is an error detecting format. Each byte of the source file is converted into two ASCII characters representing the hexadecimal value of the byte (e.g., 3F becomes "3" and "F").

The structure of the MDS file for RAM consists of a header record for a file followed by the data for that file as shown below.

Each file is preceded by a header, and the collection of files is ended with an EOF record.



All the data is split into 16-byte records, each preceded by an identifier and ended with a checksum of the record. An MDS file is roughly three times the size of the information it contains. (The MDS size times 0.35 is approximately equal to the content size.)

3-18 HXC File Conversion Utility

The RAM file-header record looks like this:



The header information in that record consists of the following the ten bytes:



Data records look like this:



Record length (data portion only) * The HP-94 ignores the load address; data is loaded sequentially. The end-of-file record (EOF) also has a specific format:



MDS File Format for ROM/EPROM Output

The structure of the MDS file for ROM differs from that for RAM in that header records for all files are collected at the beginning of the MDS file as shown below.



The directory information begins with a ROM directory-table header. The information in the directory header has the format shown below. Refer to the discussion of memory management in the HP-94

3-20 HXC File Conversion Utility

Technical Reference Manual for more information.



Directory identifier

The information in the ROM directory-entry record for each file looks like this:



Data and EOF records are the same as for RAM output.

Backup Files

The first time you run HXC, an MDS output file is created.

The next time you run HXC, the previous MDS file is renamed as a backup file (extension MBK), and a new MDS file is created.

Each subsequent time you run HXC, the previous MDS file is renamed to the backup file, and the previous backup file is lost.

The CMD file and its backup CBK file are created in an exactly analogous manner.

HXCHRSET Roman-8 Utility

HXCHRSET selects the Roman-8 character set or the standard character set for use in a Hewlett-Packard Vectra computer with a Multimode Video Adapter.



item	Description	Range
/R	literal; select Roman-8 character set	-
/S	literal: select standard character set	-

Examples

HXCHRSET /R HXCHRSET /S

Description

HXCHRSET selects the Roman-8 character set or the standard character set. It works only in a Vectra computer with a Multimode Video Adapter. Informational and self-explanatory error messages may be displayed.

If your Vectra is equiped with an HP EGA adapter, use the FONTLOAD utility to load HP8X14.FNT instead of HXCHRSET. FONTLOAD and HP8X14.FNT are both on the disc provided with the adapter.

Intended Use of HXCHRSET The HP-94 uses Roman-8 characters which are not in the standard character set of the Vectra computer. If your programs display or use the upper 128 characters of the character set, use HXCHRSET so the display in the Vectra will more closely resemble what is seen in the HP-94.

Related Commands

HXBASIC

HXCOPY File Copy Utility

The HXCOPY command transfers MDS files between the development-system computer and the HP-94.

To Copy a File From the Development-System Computer to the HP-94



To Copy a File From the HP-94 to the Development-System Computer



ltem	Description	Range
file name	literal; legal MS-DOS file name	Path, 8 chararacter filename, 3 character extension; wild cards not allowed
AUX	literal; MS-DOS device name for the first serial port (optional trailing colon)	
COMn	literal; MS-DOS device name for serial port 1-4 (optional trailing colon)	
/D	switch to display transmitted or received data during file transfer	

Examples

HXCOPY MAIN AUX HXCOPY COM2 MAIN

Description

HXCOPY is used to transfer MDS files between the development-system computer and the HP-94. The general file-transfer procedure is the same regardless of the direction in which the file is being transferred.

- 1. Start the receive operation on the receiving computer.
- 2. Start the send operation on the sending computer.

CAUTION If you do these steps in the reverse order, the receiving computer may miss some of the data.

To transfer files from the development-system computer to the HP-94

- A. Start the receive operation in the HP-94
 - 1. Turn off the HP-94, then turn it on while pressing both CLEAR and ENTER.

(This accesses the operating-system commands.)

2. Type C. < d > ENTER ENTER

(This copies from the serial port into directory $\langle d \rangle$.)

- B. Start the send operation in the development-system computer.
 - 1. Type HXCOPY <file name > AUX Enter

(This copies *<file name >* to the first serial port.)

or

2. Type HXCOPY <file name > AUX /D Enter

(The /D causes the the file to be displayed as it is sent.)

To transfer files from the HP-94 to the development-system computer

- A. Start the receive operation in the development-system computer.
 - 1. Type HXCOPY AUX <file name > Enter

(This copies from the first serial port into <file name>.)

or

2. Type HXCOPY AUX <file name > /D Enter

5-2 HXCOPY File Copy Utility

(The /D causes the file to be displayed as it is received.)

- B. Start the send operation in the HP-94.
 - 1. Turn off the HP-94, then turn it on while pressing both CLEAR and ENTER.

(This accesses the operating-system commands.)

2. Type C<d>: <file name > ENTER ENTER

(This copies $\langle file name \rangle$ in directory $\langle d \rangle$ to the serial port.)

NOTE Anywhere AUX was used, you can substitute COMn where n is 1, 2, 3, or 4.

Things to Remember About the Development-System Computer

You must initialize and configure the serial port in the development-system computer before you can transfer files. Do this by executing the HXMODE command (see chapter 7 of this manual). (You can include HXMODE in your AUTOEXEC. BAT file so it will be executed automatically when the development-system computer is turned on. HXMODE needs to be executed only once: whenever the development system is turned off or re-booted.

You must use HXC to convert your file(s) to MDS format before it can be transferred.

Halt HXCOPY by pressing CTRL C or CTRL Break or CTRL ScrLck).

When running HXCOPY, the use of a file-name extension is optional. If you do not supply an extension, HXCOPY will use MDS automatically.

Things to Remember About the HP-94

< d > represents the directory number in the HP-94:

0 for main memory. 1 for an HP 82411A 40K RAM Card. 1, 2, 3, or 4 for an HP 82412A ROM/EPROM Card.

Press SHIFT SPACE for the colon character (:) between the directory number and file name.

Refer to chapter 8 "HP-94 Operating Commands" for more information about the C command.

About HXCOPY

HXCOPY is a development tool only. It should not be used as the primary method of data communications in an application program for the following reasons:

MDS is an error-detecting but not an error-correcting format. There is no provision for automatically retransmitting records that were not transmitted properly. Therefore, important data might be lost.

- HXCOPY receives files in MDS format only. Most host computers will not have utilities for converting MDS files into their decoded form.
- It is difficult to synchronize the C command on the HP-94 with HXCOPY running on a remote computer.
- HP-94 operating-system commands are cryptic and can be difficult to use.

Messages

All informational or error messages you may see are self-explanatory.

Related Commands

HXC, HXMODE

HXMODE Handshaking Utility

The HXMODE utility provides XON/XOFF handshaking between HXCOPY in the developmentsystem computer and the HP-94.



item	Description	Range)
/X	switch for enabling XON/XOFF	—
/N	switch for disabling XON/XOFF	_
AUX	literal; MS-DOS device name for the first serial port; (optional trailing colon)	_
COMn	literal; MS-DOS device name for serial port 1-4 (optional trailing colon)	_
baud rate	integer	9600, 4800, 2400, 1200, 600, 300, 150
data format	literal	7S, 7ES, 7OS, 8S, 8ES, 8OS, 7SS, 7ESS, 7OSS, 8SS, 8ESS, 8OSS
receive buffer size	integer	1 through 64 paragraphs of 16 bytes each

Examples

HXMODE HXMODE /X AUX 9600 7ES HXMODE COM2 1200

Description

The main function of HXMODE is to provide XON/XOFF handshaking between the developmentsystem computer and the HP-94. It is used in conjunction with the HXCOPY utility for file transfers to and from the HP-94. HXMODE also defines the serial port configuration.

HXMODE has five optional parameters. Their default values are:

ltem	Default
XON/XOFF	enabled
serial port	AUX
baud rate	9600
data format	7ES
receive buffer size	4 paragraphs

The HP-94 default configuration is the same.

The /X and /N switches indicate whether or not to use XON/XOFF handshaking. /X means use XON/XOFF, and /N means do not use XON/XOFF. When the serial port is opened by HXCOPY, a single XON is sent. When the receive buffer is 3/4 full (48 bytes for the default buffer size), a single XOFF is sent. An XON is sent when the receive buffer is emptied to the half-full point.

The serial port can be specified by AUX for the first serial port or COM1 through COM4 for the first through the fourth serial ports.

The data format acommodates 7- or 8-bit data, even, odd, or no parity (E, O, or nothing), and one or two stop bits (S or SS).

The receive buffer is specified in paragraphs (blocks of 16 bytes).

Things to Remember

HXMODE used alone is equivalent to HXMODE /X AUX 9600 7ES 4.

HXMODE does not operate as an MS-DOS device driver; it leaves itself resident in memory just as does the MS-DOS PRINT command. HXMODE indicates with a message that it has been installed the first time it is executed and subsequent times only when the buffer size increases. To remove it from memory, reboot the development-system computer by pressing CTRL, Att, and delete it from the AUTOEXEC. BAT file, if necessary.)

XON/XOFF handshaking for HXCOPY can be disabled be re-executing HXMODE with the /N switch.

6-2 HXMODE Handshaking Utility

HXMODE needs to be executed only once whenever the development system computer is turned off or rebooted.

The HXMODE utility program and the MS-DOS MODE command are independent of each other. When HXMODE is executed, it does not immediately change the serial port configuration. Instead, it waits until HXCOPY is transferring a file to or from the HP-94. It then changes the serial port configuration, performs handshaking during the file transfer, and restores the port configuration to its original state. Therefore, in order to change the serial port configuration for other uses of the serial port (including for HXBASIC), you must still use the MS-DOS MODE command.

Messages

All informational and error messages you may see are self-explanatory.

Related Commands

HXCOPY

Operating-System Commands

Introduction

The HP-94 operating system controls the hardware resources of the HP-94. The operating-system commands provide file management functions and limited self-test capabilities.

To Access the Operating Commands

To access the operating commands, turn on the computer while pressing the ENTER and CLEAR keys at the same time. The following message is displayed while the operating system waits for a command:

Copr. HP 1985 Vxx

where Vx.xx specifies the version number of the operating system.

Things to Remember

- The HP-94 will turn itself off automatically if no keys have been pressed and no data has been received at the serial port after a period of two minutes.
- Directories are numbered as follows:

0 = main memory (64K-, 128K-, or 256K-bytes of RAM depending on the model and configuration).

1 = HP 82411A 40K RAM Card.

1, 2, 3, or 4 = HP 82412A ROM Card.

5 = operating-system directory -- cannot be erased or initialized.

- HP-94 file names are 1 4 alphanumeric characters; the first character of the name must be an alpha character.
- To change from alpha characters to numeric characters on the HP-94 keyboard, press the SHIFT key. The keys remain shifted until you press the SHIFT key again.

When the orange alpha keys are active, the cursor is a blinking black square.

Operating-System Commands 7-1

When the white numeric keys are active, the cursor is a blinking underline.

(The ..., -..., CLEAR, and ENTER keys are active in both character sets.)

To type a colon (:), press the SPACE key.

How to Handle Errors

Refer to appendix A for information on how to handle errors.

The Commands

The following table lists the operating-system commands:

Command	Function
B	Change baud rate or data format
	of the serial port
С	Copy file
D	Directory list
Е	Erase file
I	Initialize a directory
K	Keyboard self test
L	LCD self test
M	Memory test
S	Start an application program
Т	Set time and date

A detailed description of each command follows.

7-2 Operating-System Commands

The B command sets the baud rate and data format for the serial port.



Item	Description	Range
baud-rate specifier	integer - refer to next table	1 - 7
data-format specifier	integer - refer to next table	0 - 15

Example

B2.5 - sets the baud rate to 4800 with a word length of 8, 1 stop bit, and no parity.

Description

The B command sets the baud rate and data format for the serial port. The baud rate and data format specifiers can have any of the values shown in the following tables:

Specifier	Baud Rate
1 (default)	9600
2	4800
3	2400
4	1200
5	600
6	300
7	150
[

Baud-Rate Specifier

Specifier	Word Length (bits)	Stop Bits	Parity
0	7	1	none
1	8	1	none
2	7	1	odd
3	8	1	odd
4	7	1	none
5	8	1	none
6	7	1	even
7	8	1	even
8	7	2	none
9	· 8	2	none
10	7	2	odd
11	8	2	odd
12	7	2	none
13	8	2	none
14	7	2	even
15	8	2	even

Data-Format Specifier

NOTE In the case of duplicate specifiers, either specifier may be used.

Things to Remember

For 7-bit data only:

- Specifiers 1 and 5 are equivalent to a word length of 7, 1 stop bit, and 0's parity.
- Specifiers 8 and 12 are equivalent to a word length of 7, 1 stop bit, and 1's parity.

The default configuration of the serial port when the HP-94 is turned on is 9600 baud, 7-bit data, even parity, 1 stop bit (setting B1.6). XON/XOFF handshaking is performed automatically.

The serial-port configuration set by the B command is in effect only when the C command is being used. BASIC programs automatically set the configuration to the default, whether or not the B command changed it.

To change the configuration in a BASIC program, use the SYRS command.

Related Commands

None.

7-4 Operating-System Commands



The C command copies the specified MDS file into the specified directory

Examples

C.0	- reads the incoming files from the serial port, and places them into directory 0 (main directory).
C*	- copies all files from the first non-empty directory to the serial port (and from there to the development-system computer).
C1:*	- copies all files in directory 1 to the serial port.
C1:TEST.0	- copies the file TEST in directory 1 to a file named TEST in directory 0.

Description

The general syntax for the C command is C<source>.<destination>.

If < source> is omitted, the serial port is used, and files are copied into the HP-94.

If <destination> is omitted, the serial port is used, and files are copied out of the HP-94.

Operating-System Commands 7-5

If neither is omitted, the copy operation occurs between one directory and another within the HP-94.

If the file to be copied already exists in the destination directory, the existing file is replaced by the copied file.

If the number of the directory containing a file to be transferred is omitted, a sequential search is made from directory 0 through the last directory until a file having the same name is found. That file is then copied to the designated destination.

An * indicates all files in the specified directory.

When the confirmation message Ok ? is displayed, copying begins when the ENTER key is pressed. If any key other than ENTER is pressed, the file(s) will not be copied.

While copying is in progress, dots will repeat at the end of each line indicating that the file is being copied.

Related Commands

D, E, I, M

The D command displays information concerning the files in the designated directory.



ltem	Description	Range
directory number	integer	0 through 5 (refer to the S command)

Examples

D	- displays information about all the directories resident in the HP-94.	
DO	- displays information about all of the files resident in directory 0 (the main direc- tory).	
D5	- displays information about the files resident in directory 5 including the SYBI snd SYOS files.	

Description

For information concerning directory n, type Dn.

The display format for the D command is: Dirn L aaaa bbbb <file name(s) > Fcccc dddd eeee Free where *n*=the directory number L = the directory type; M (main memory) A (40K-byte RAM card) O (ROM/EPROM card) aaaa = the number of 16-byte paragraphs (hex) bbbb = the address of the starting segment of the directory (hex) F = the file type; A (assembly-language program) B (BASIC program) D (data file) H (handler) cccc = the number of 16-byte paragraphs in the file (hex) dddd = the address of the starting segment of the file (hex) eeee = the free space in the directory in 16-byte paragraphs (hex) An asterisk (*) to the left of a file name indicates a checksum error has occurred; the file may be corrupted.

A maximum of four lines of information can be displayed at one time.

For directories, the directory information and the available memory is displayed.

For files within a directory, the file information and the available memory is displayed.

Press the **ENTER** key to scroll the four lines up out of the display and show the next four lines of information.

Press the ... key to scroll the top line up out of the display and show the next line of information.

Press any other key to end the information display.

If a file is corrupted, you may:

- Erase the file, or
- Upload the file to the development-system computer, or

7-8 Operating-System Commands

■ Use the file "as is."

You will probably want to reload program files or correct data files.

You can use corrupted files normally (programs will run, and data files can be read), but be aware that you risk bad results when running such programs or using bad data.

Use the M command to display only the corrupted files.

The asterisk will not appear in the file name field of the directory display if a checksum error 217 is displayed for an HP 82412A ROM/EPROM card. The fact that the error was reported indicates the contents of the ROM or EPROM has changed; the ROM or EPROM must be replaced.

Related Commands

C, E, I, M

.

The E command erases the specified file from the specified directory.



ltem	Description	Range	
directory number	integer	0 through 5 (refer to the S command)	
filename	any legal filename	alphanumeric	

Examples

E1:TEST - erases the file named TEST in directory 1.

E1:* - erases all the files in directory 1.

Description

If the directory is omitted, a sequential search is made for the specified file starting from directory 0. The first file with the specified filename is erased.

If the specified file does not exist, an error is reported.

An * indicates all files in the specified directory.

When the confirmation message Ok ? is displayed, the file is erased when the ENTER key is pressed. If any other key is pressed, the file is not erased.

Specifying the * will erase all the files in the specified directory or will erase all the files in the first non-empty directory if the directory number is omitted. You must confirm each file to be erased by responding to the confirmation message.

Related Commands

C, D, M

7-10 Operating-System Commands
The I command initializes the specified directory.



Item	Description	Range
directory number	integer	0 or 1

Example

IO - initializes directory 0 (main memory).

Descriptions

Before any files can be loaded into a new HP-94, the main memory must be initialized with IO.

Before any file can be loaded into a new HP 82411A RAM Card, the card must be initialized with I1.

Main memory must be initialized with I0 when error 212 occurs.

An HP 82411A RAM Card must be initialized with I1 when error 213 occurs.

When the confirmation message Ok ? appears in the display, initialization starts when the **ENTER** key is pressed. If any other key is pressed, the directory will not be initialized, and the system waits for a new command.

CAUTION Initializing a directory will destroy all the files in that directory.

Related Commands

None.

The K command causes the HP-94 to check its keyboard.



Example

K

Description

If the keyboard is working correctly, the corresponding letter, number, or symbol is displayed when an HP-94 key is pressed. The self-check is terminated when the **ENTER** key is pressed.

Related Commands

None.

7-12 Operating-System Commands

The L command causes the HP-94 to check its display.



Example

 \mathbf{L}

Description

The first character in the Roman-8 character set is repeated across a line of the display. Every 1/2 second, the next character in the character-set sequence appears across a line of the display. The check ends when all the characters in the set (except control codes and user-defined characters) have been displayed. The check stops and is ended when any key is pressed.

Related Commands

None.

L

The M command indicates which files had checksum errors when the HP-94 was turned on.



Item	Description	Range
directory number	integer	0 through 5 (see S command)

Examples

M	-the same as the D command. It shows which directories are installed in the HP-94.
MO	-shows which files have been corrupted in directory 0.

Description

The M command uses the same display format as the D command except that the only information displayed is for corrupted files. Refer to the D command for details.

If checksum error 217 occurs when an HP 82412A ROM/EPROM Card is installed, the asterisk will not appear in the filename field of the directory display. The fact that the error appeared indicates that the contents of the ROM or EPROM changed; the ROM or EPROM must be replaced.

Related Commands

C, D, I

7-14 Operating-System Commands

The S command starts the specified application program.



ltem	Description	Range
directory number	integer	0 through 5 (see note
filename	any legal filename	alphanumeric

NOTE Directories 1, 2, 3, and 4 are available when an HP 82412A ROM/EPROM card is installed.

Directory 1 is also available when an HP 82411A 40K RAM card is installed.

Examples

S	-starting with directory 0 searches for and runs the first program named MAIN.
SCASH	-starting with directory 0 searches for and runs the first program named CASH.
S1:MONY	-runs the program MONY in directory 1.

Description

Whenever the S command is used to start an application program, the program starts from its beginning.

When the directory number is omitted, a sequential search is made, starting with directory 0 but not including directory 5.

If the filename is omitted, the program named MAIN is searched for and run.

If the specified program does not exist, an error is reported.

Operating-System Commands 7-15

...S

If the specified file is the file containing the BASIC interpreter (SYBI), it will not be executed.

If the specified file is the file containing the operating system (SYOS), the HP-94 will automatically turn off.

CAUTION Do not try to run SYFT (user-defined font). Unpredictable, and possibly harmful side effects will occur.

Related Commands

D

The T command displays the real-time clock and allows you to change the current time and date.



Examples

т

Description

Pressing T ENTER displays the current date and time and waits for you to input a new date and time. If the existing date and time are accurate, press ENTER again to leave them unchanged. If not, type in a new date and time, and press ENTER.

When entering a date and time, all items must have two digits, be separated by a period, and be entered in the following order:

MM.DD.YY.hh.mm.ss (month.day.year.hour.minutes.seconds)

The HP-94 does not check the validity of the input data except to verify that the correct number of characters was used.

Related Commands

None.

SYBD HP-94 BASIC Debugger

Introduction

Since your development-system computer does not support all of the keywords supported by the HP-94, you will have to do the final debugging of your application programs in the HP-94. The SYBD debugger program provides this capability.

Using SYBD

The debugger program is contained in a file called SYBD.MDS on your system disc. To use it do the following:

- 1. Using HXCOPY (described elsewhere in this manual), load SYBD.MDS and your application program into the HP-94.
- 2. Start executing the application program. The debugger displays the following prompt and waits for a debug command input.



where mmmmm is the line number that is going to be executed, and

pppp is the name of the program.

The Debug Commands

The following table lists the debug commands, each of which are described in detail later in this chapter.

Command	Function
A	Display the starting address of the variable area
В	Set a breakpoint
Е	Exit from debugger, and enter command mode
M	Read or modify memory
R	Start or restart the execution of a program
U	Clear a breakpoint
v	Read or modify variable-area memory
ENTER	Single step execution of the program
CLEAR	Clear the field currently being edited
←-	Delete the character to the left of the cursor

Things to Remember

- The HP-94 does not turn off automatically while waiting for a command at a breakpoint.
- All commands and hexadecimal characters can be entered without using the SHIFT key.
- It is not possible to set a breakpoint in a program in ROM since the breakpoint commands alter the line-number field of the program.
- The ← and CLEAR keys are used to edit program entries.
- Invalid characters will be ignored.
- Line numbers are in decimal; segment and offset values are in hexadecimal.

The A command displays the variable-area starting address of the current program.

A

Description

When prompted for a debug command, press A. The variable-area starting address of the current program is displayed as follows:

ssss:ffff

where ssss is the segment address and

ffff is the offset.

The B command sets a breakpoint at a given line in the BASIC program.



ltem	Description	Range
line number	integer	0 - 32767

Description

When prompted for a debug command, press B [nnnnn] ENTER to set a breakpoint in line nnnnn.

Pressing **B** ENTER without specifying a line number sets a breakpoint at the current line in the program.

To set breakpoints in a subprogram, set a breakpoint in the line calling the subprogram. Run the program. When the breakpoint is reached, press **ENTER** to get to the first line of the subprogram, and then set breakpoints in the subprogram.

Breakpoints cannot be set in EPROM/ROM.

The E command exits from the debug program and returns to the command mode.

Description

When prompted for a debug command, press E. The E command then prompts with Ok ? to verify the intention to exit.

Press ENTER to exit SYBD and return to command mode.

Press any key other than ENTER to abort the E command and remain in the debug program.

The M command examines or modifies memory.



ltem	Description	Range
segment	integer	0
offset	integer	0 - FFFFh

Description

When prompted for a debug command, press [M].

- The first time you use the M command, 0000:0000 is displayed.
- If you have used M or V before, the previous segment address and offset are displayed.

If the displayed address is the memory location you wish to see, press **ENTER**.

If you wish to see a different memory location, type [ssss:ffff] where ssss is the segment address and ffff is the offset of the memory location you wish to see, and press **ENTER**.

In either case, the display changes to the following:

ssss:ffff xx	
--------------	--

where xx is a two-digit hexadecimal number representing the contents of the memory location.

You now have four options. Refer to the table below, and press the indicated key(s):

8-6 SYBD HP-94 BASIC Debugger

M

Desired Action	Keys
Advance to next address	ENTER
Change memory contents and advance to next address	yy ENTER
End M command	
Change memory contents, and end M command	уу .

Invalid inputs are ignored.

When the offset (ffff) advances from FFFFh to 0000h, 1000h is added to the segment address (ssss).

To enter the :, press SPACE or ...

The character immediately to the left of the cursor can be deleted by pressing the \leftarrow key.

SYBD HP-94 BASIC Debugger 8-7

The R command starts or restarts the BASIC program.



Item	Description	Range
line number	integer	0 - 32767

Description

When prompted for a debug command, press [R] [nnnnn] [ENTER]. The BASIC program starts running at line nnnnn. A breakpoint at line nnnnn is ignored.

If no line number is specified, the program starts running at the current line.

8-8 SYBD HP-94 BASIC Debugger

R

The U command clears a breakpoint in the BASIC program.



ltem	Description	Range
line number	integer	0 - 32767

Description

When prompted for a debug command, press \boxed{U} [*nnnn*] $\boxed{\text{ENTER}}$. The breakpoint at line *nnnnn* is cleared.

If no line number is specified, the breakpoint at the current line is cleared.

The V command examines or modifies variable-area memory.



ltem	Description	Range
segment	integer	0 - FFFFh
offset	integer	0 - FFFFh

Description

When prompted for a debug command, press [V], and enter the address of the variable as shown in the BMP file produced by HXC for that BASIC program.

The variable-area starting address can be displayed using the A command.

The operation of the V command is identical to the M command except that V examines variablearea memory. The ENTER command executes the BASIC program line by line.



Description

When prompted for a debug command, press **ENTER**. The current line is executed, and the program advances to the next line and stops.

[CLEAR]

The CLEAR command clears a group of one or more characters.



Description

Before you press the **ENTER** key, any character or group of characters you have entered may be erased by pressing the **CLEAR** key.

.

.

The \leftarrow command moves the cursor one space to the left and removes one character.

$$\bigcirc$$

Description

Before you press the $\boxed{\text{ENTER}}$ key, any character or group of characters you have entered may be erased one-at-a-time by pressing the $\boxed{-}$ key.

Error Handling

Introduction

Errors may result from hardware or software failures. The following paragraphs describe the errors and how they are checked and reported.

Non-numeric errors are generated by the BASIC interpreter in the HP-94; numeric errors are generated by the operating system in the HP-94.

You should verify an error by repeating the procedure which caused it to be reported before taking further action.

Appendix B describes diagnostic tests you can use to further verify hardware failures.

Hardware Errors

Before the HP-94 completes the turn-off operation, it computes checksums for important memory areas and a checksum for each file in its memory. These checksums are recomputed when the HP-94 is turned on and compared to their original values.

Directory Problems: If the "off/on" checksums for a directory area do not match, a hardware failure has occurred in the HP-94. Error message 212 or 213 is displayed indicating which block of memory has been corrupted. Use the I command to reinitialize the directory (I0 if only main memory is present; I1 if a 40K RAM card is present) before verifying the error.

Scratch Area Problems: If the "off/on" checksums for a scratch area do not match, a hardware failure has occurred in the HP-94. Error message 214 or 215 is displayed.

File Problems: If the "off/on" checksums for a file do not match, error 216 is displayed if the file is in main memory or error 217 is displayed if the file is in a RAM or a ROM card. Use the D (directory) or M (memory) commands to determine which files were corrupted so you can take corrective action.

Additional Turn-On Errors: If at turn on you hear a beep and the HP-94 will not enter the command mode, or you hear a beep and the HP-94 displays an error message consisting of an 8-digit hexadecimal number, you have main-memory failure.

Software Errors

Non-numeric and numeric errors resulting from software problems are reported as follows:

Non-Numeric Errors: Non-numeric errors always cause the BASIC-program execution to halt and control to return to the operating system. An error message

Error MM LLLLL PPPP

is displayed where

MM is the error message,

LLLLL is the BASIC program line number, and

PPPP is the name of the BASIC program or subprogram in which the error occurred.

The following table lists the non-numeric errors:

Non-Numeric Errors From The HP-94 BASIC Interpreter

Message	Meaning
AR	Array subscript error
BM	BASIC interpreter malfunction
BR	Branch destination error
CN	Data conversion error
CO	Conversion overflow
DO	Decimal overflow
DT	Data error
EP	Missing END statement
FN	Illegal DEF FN statement
IL	lllegal argument
IR	Insufficient RAM
IS	Illegal statement
LN	Nonexistent line
MO	Memory overflow
NF	Program not found
RT	RETURN or SYRT error
SY	Syntax error
TY	Data type mismatch
UM	Unmatched number of arguments

A-2 Error Handling

Numeric Errors: If no alternate behavior for error processing is specified, numeric errors cause the BASIC program execution to halt. An error message

Error NNN LLLLL PPPP

is displayed where

NNN is the error number,

LLLLL is the BASIC program line number, and

PPPP is the name of the BASIC program or subprogram in which the error occurred.

Message	Meaning
100	BASIC interpreter not found *
101	illegal parameter
102	Directory does not exist
103	File not found
104	Too many files
105	Channel not open
106	Channel already open
107	File already open
108	File already exists
109	Read-only access
110	Access restricted
111	No room for file
112	No room to expand file
113	No room for scratch area
114	Scratch area does not exist
115	Short record detected **
116	Terminate character detected **
117	End of data *
118	Timeout
119	Power switch pressed
200	Low main battery voltage
201	Receive buffer overflow
202	Parity error
203	Overrun error
204	Parity and overrun error
205	Framing error
206	Framing and parity error
207	Framing and overrun error
208	Framing, overrun, and parity error
209	Invalid MDS file received **
210	Low backup battery voltage - main memory *
211	Low backup battery voltage - memory board or 40K RAM card *
212	Checksum error - main memory directory table*
213	Checksum error - 40K RAM or ROM/EPROM card directory table*
214	Checksum error - operating system scratch area *
215	Checksum error - main memory free area *
216	Checksum error - main memory file *
217	Checksum error - 40K RAM or ROM/EPROM card file *
218	Lost connection while transmitting
219	Operating system stack altered *
	* Only reported when the HP-94 is turned on.
	** Not reported to BASIC programs.

Numeric Errors From the HP-94 Operating System

Things To Remember

SYER has no affect on non-numeric errors.

The SYER statement can be used in the BASIC program to ignore errors and allow program execution to continue uninterrupted.

If a numeric error takes place after SYER with specifier 0 is executed, the error number is assigned to the error-number variable, allowing programmatic error handling or error trapping.

The program name, line number, channel number, and the I/O length of the most recent error can be determined with the SYIN statement.

Diagnostic Tests

Introduction

There are several diagnostic tests you can run to determine the functionality of your HP-94 before sending it in for repair. These tests are not exhaustive; however, they can give you an indication of which parts are defective.

Before taking any corrective action, you should:

- Make sure your development-system computer is functioning properly.
- Make sure all cables are installed and configured correctly. (Refer to the Getting Started Guide, page 28 for further helpful information.)
- Always verify the existence of the problem by repeating the procedure that caused it to be reported or to manifest itself.
- If you are uncertain whether the problem is with the HP-94 or elsewhere, try repeating the procedure with another HP-94.

You Can't Download Data to the HP-94

This problem might be caused by a hardware failure in the RS-232-C serial port in the HP-94 or in the development-system computer, by a failure in the interconnection cable, or it might be caused by your software. To isolate the problem, do the following:

- 1. Using the procedures described elsewhere in this manual, download the SYBD program into the HP-94.
 - If the download is successful, your software is at fault. Erase SYBD, and examine your program for errors.
 - If the download is not successful, go to step 2.
- 2. Replace the interconnection cable. and download the SYBD program into the HP-94.
 - If the download is successful, your cable is faulty. Replace it.
 - If the download is not successful, your cable is not faulty; go to step 3.
- 3. Replace the HP-94, and download the SYBD program into the HP-94.
 - If the download is successful, the serial port in your HP-94 is probably bad. Send the HP-94 in for repair.

If the download is not successful, you may have a hardware failure in your HP-94 or you may have a failure in your development-system computer.

If you have another development-system computer available, you can perform another step to isolate the problem by using it in this test.

Some Parts of the Displayed Characters Are Missing

To verify the problem, do the following:

- 1. Enter the command mode by turning the HP-94 off, and then turning it on while holding down the CLEAR and ENTER keys.
- 2. Press L ENTER to run the display test.

A complete character font scrolls across the display from bottom to top. Stop the display test by pressing ENTER. (You will have to restart the test to get the characters moving again.)

Examine each character carefully. If any columns or rows of dots are missing, the display needs repair.

The Bar-Code Wand Doesn't Work

NOTE This test assumes that you are knowledgeable about the operation of the bar-code wand you are using, and that the wand is reasonably functional; i.e., there are no obvious signs of damage, the lens is clean, and the wand is connected to the HP-94 correctly.

If the bar-code wand doesn't work, it may be your application program, the bar-code wand, or the HP-94 (or any combination of these) that is at fault.

NOTE The following test applies to the Hewlett-Packard Smart Wands only. If you are using some other make of wand, you will have to adapt the test to your wand's characteristics.

To isolate the problem, do the following:

1. Using HXBASIC, create the following program called "MAIN":

```
10 DIM barcode$255
20 OPEN #2,"HNBC"
30 barcode$=""
40 GET #2 barcode$
50 IF barcode$="" THEN CLOSE#2:END
60 PRINT %HOME;barcode$
70 GOTO 30
```

2. Using HXC and HXCOPY, as described elsewhere in this manual, load MAIN.BAS, and HNBC.EXE into the HP-94. (HNBC.EXE is on your development-system disc.)

B-2 Diagnostic Tests

- 3. Turn the HP-94 off and then on.
 - If ready 12.3 is displayed, the HP-94 is good. Go to step 4.
 - If the message is not displayed, go to step 5.
- 4. Scan some bar code.
 - If the bar code is read and displayed properly, the wand and bar-code port are good. Examine your program for errors.
 - If the bar code is not read and displayed properly, the wand is probably bad. Replace it, or consult your Hewlett-Packard representative regarding service for it.
- 5. Scan the HP-94 default-configuration bar code, then turn the HP-94 off and then on. (You should have received this bar code with your Hewlett-Packard Smart Wand.)
 - If the message is displayed, the wand is good.
 - If the message is not displayed, go to step 6.
- 6. Replace the wand, then turn the HP-94 off and then on.
 - If the message is displayed, your wand is bad.
 - If the message is not displayed, reinstall your wand, and go to step 7.
- 7. Replace the HP-94, then turn it on, and repeat steps 2 through 6. If the bar code is read and displayed properly, the bar-code port in your HP-94 is probably bad. Send the HP-94 in for service.



Printed in U.S.A. 12/86