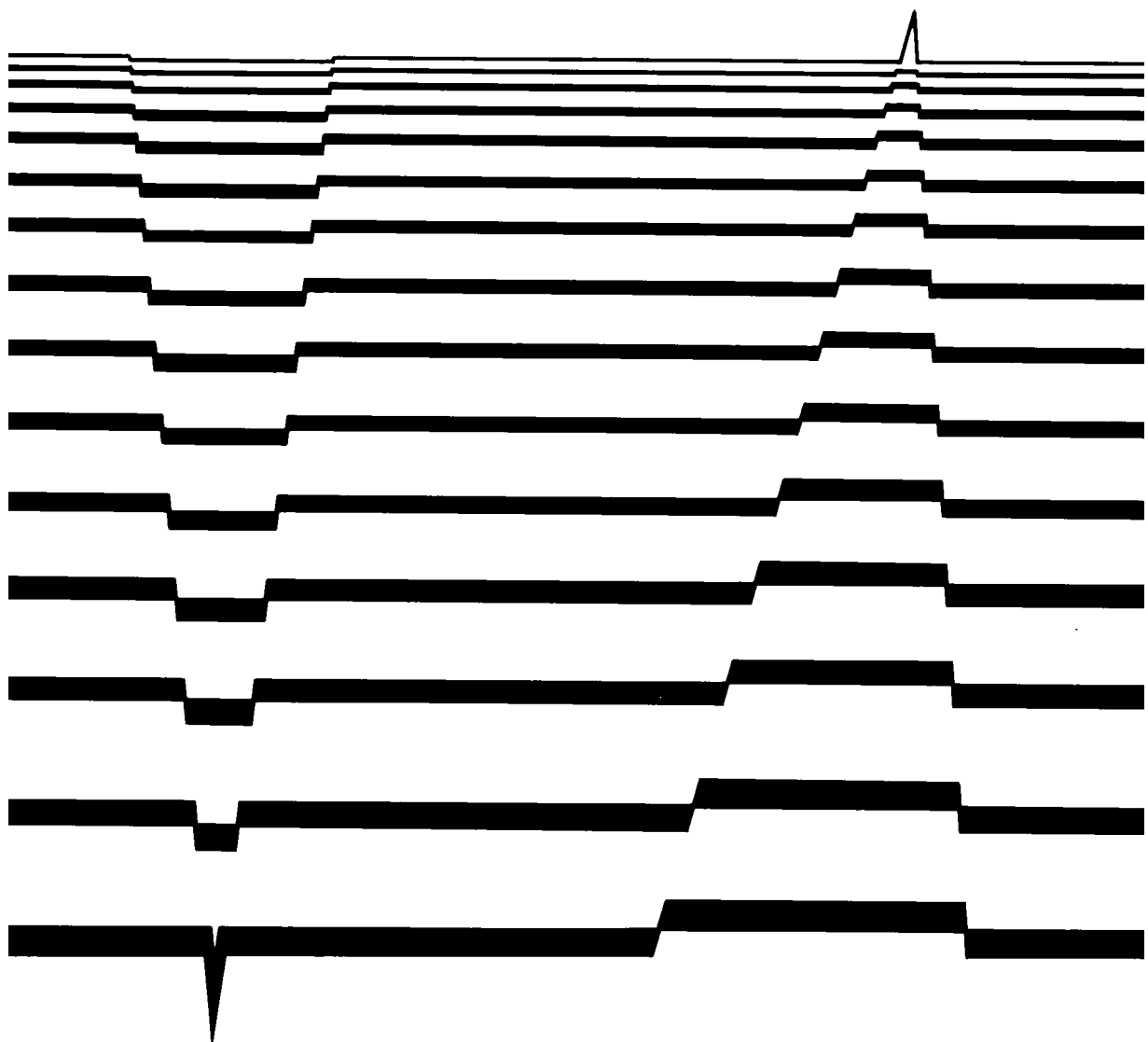


# YAMAHA

**MSX<sub>2</sub>**

## YAMAHA MSX CP/M<sup>®</sup>

### OPERATION MANUAL



CP/M is a registered trademark of Digital Research.

**MSX**<sub>2</sub> is a trademark of ASCII Corporation.

# **DIGITAL RESEARCH (JAPAN) INC.**

## **END USER PROGRAM LICENSE AGREEMENT**

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THE DISKETTE PACKAGE.

Title to the media on which the program is recorded and to the documentation in support thereof is transferred to you, but title to the program, and all subsequent copies of the program, regardless of the form or media in or on which the original and other copies may subsequently exist, is retained by DRJ or AUTHOR. This license is not a sale of the original or any subsequent copy. You assume responsibility for the selection of the program to achieve your intended results, and for the installation, use and results obtained from the program.

### **1. DEFINITIONS — In this License Agreement, the terms:**

- A. DRJ means DIGITAL RESEARCH (JAPAN) INC., NCR Shibuya Building, 14th F1., 16-16 Nampoedai-cho, Shibuya-ku, Tokyo 150, Japan, owner of the copyright in, or authorized licensor of, the program.
- B. Machine means the single microcomputer on which you use the program. Multiple CPU systems require additional licenses.
- C. AUTHOR means any third party author and owner of the copyright in this program.
- D. Run-time Library means the set of copyrighted language subroutines, provided with each language compiler, a portion of which must normally be linked to and become part of your program for that program to run on a computer.

### **2. LICENSE**

You may:

- a) use the program on a single machine; and
- b) copy the program into any machine readable form for back-up purposes only in support of such use. (Certain programs, however, may include mechanisms to limit or inhibit copying. They are marked "copy protected."); and
- c) modify the program and/or merge the program into another program for your use on the single machine (Any portion of this program merged into another program will continue to be subject to the terms and conditions of this Agreement); and,
- d) transfer the program and license to another party if you notify DRJ of the name and address of the other party and the other party agrees to a) accept the terms and conditions of this Agreement, and b) pay the then current transfer fee. If you transfer the program, you must at the same time either transfer all copies, including the original, whether in printed or machine-readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the program contained or merged into other programs.

You must reproduce and include the copyright notice on any copy, modification or portion merged into another program.

You may not use, copy, modify, or otherwise transfer the program, or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this Agreement.

You may not disassemble, decompile, or otherwise reverse engineer the program.

You may not copy documentation or other printed materials.

If you transfer possession of any copy, modification or merged portion of the program to another party, your license is automatically terminated.

### **3. TERM**

The license is effective until terminated. You may terminate it at any other time by destroying the program together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement, or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the program together with all copies, modifications and merged portions in any form.

### **4. COMPOSITE PROGRAMS**

As an exception to Paragraph 2, you are granted the right to link portions of any included Run-time Library in your developed programs, called Composite Programs, and to use, distribute and license such Composite Programs to third parties without payment of any further license fee. You shall, however, include in such Composite Program, and on the exterior label of every diskette, a copyright notice in this form: "Portions of this program, Copyright (C) 1986 (or other appropriate year) Digital Reserach Inc. (and/or AUTHOR as appropriate). All Rights Reserved. "In cases where such Composite Program is contained in Read-Only-Memory (ROM) chips, a copyright notice in the form listed above, must be displayed on the exterior of the chip and internally in the chip (in ASCII literal form). As an express condition to the use of the Run-time Library, you agree to indemnify and hold DRJ harmless from all claims by you and third parties arising out of the use of Composite Programs.

### **5. LIMITED WARRANTY AND DISCLAIMER OF WARRANTY**

DRJ warrants the media on which the program is furnished to be free from defects in materials and workmanship under normal use for ninety (90) days from the date of delivery to you as evidenced by a copy of your receipt.

DRJ warrants that each program that is designated by DRJ as warranted in its program specifications, supplied with the program, will conform to such specifications provided that the program, will conform to such specifications provided that the program is properly used on a machine for which it was designed. If you believe that there is a defect in a warranted program such that it does not meet its specifications, you must notify DRJ through your distributor within the warranty period and in the manner set forth in the program specifications.

ALL OTHER PROGRAMS ARE PROVIDED "AS IS" AND NEITHER DRJ, ITS DISTRIBUTORS OR DEALERS, NOR AUTHOR MAKE ANY WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU (AND NOT DRJ OR AUTHOR) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Neither DRJ, its Distributors or Dealers, nor AUTHOR warrant that the functions contained in any program will meet your requirements or that the operation of the program will be uninterrupted or error free or that program defects will be corrected.

## **6. GENERAL**

**You may not sublicense, rent or lease this program. Any attempt to sublicense, rent or lease or, except as expressly provided for in this Agreement, to transfer any of the rights, duties or obligations hereunder is void.**

**AUTHOR is an intended third party beneficiary of all of the exclusions, limitations, restrictions and conditions imposed in this Agreement and shall be entitled to enforce the same by direct action.**

**This Agreement will be construed under the laws of Japan.**



# CONTENTS

## Chapter 1. UNIQUE COMMANDS TO YAMAHA MSX CP/M

FEATURES .....	2
COMMANDS & OPERATION .....	2

## Chapter 2. REFERENCE

INTRODUCTION.....	8
CONTROL KEYS .....	9
ASM.COM .....	10
DDT.COM .....	11
DIR .....	14
DUMP.COM.....	15
ED.COM .....	16
ERA .....	17
LOAD.COM .....	18
MOVCPM.COM.....	19
PIP.COM.....	20
REN .....	22
SAVE.COM.....	23
STAT.COM.....	24
SUBMIT.COM.....	27
TYPE.....	28
XSUB.COM .....	29

## APPENDIX

APPENDIX A.....	32
APPENDIX B.....	33
APPENDIX C.....	35
APPENDIX D .....	36
APPENDIX E.....	37
APPENDIX F .....	39





## **Chapter 1**

---

# **UNIQUE COMMANDS TO YAMAHA MSX CP/M**

## FEATURES

The Yamaha MSX CP/M ver. 2.2™ is a 56K CP/M software system for network access, including the network SHELL. Floppy-disk based CP/M software can be used with any MSX personal computer having a memory capacity of 64KBytes or more.

CP/M software can be driven not only in disk-based form, but in a ROM base as well. Using an MSX2 personal computer with 128KBytes or more, even without a floppy disk unit, CP/M software is driven, the network is accessed, and the files of CP/M software can be manipulated.

Since the disk is given priority over the ROM, if a floppy disk unit is attached to the MSX computer, operation cannot be performed with ROM-based CP/M software. In this case, insert the CP/M software disk in the floppy disk unit and perform operations through disk-based CP/M software.

## COMMANDS & OPERATION

- **Version**  
Ver. 2.2 (Network correspondence, TPA = 56Kbytes)
- **Supplied Device**  
3.5-inch 2DD floppy disk (720Kbytes), or 32-Kbyte ROM (YIS503IIIR only; includes 16-Kbyte CP/M)
- **Memory Size**  
64Kbytes min. (Floppy-disk based; if a computer with more than 64Kbytes is used, the extra is used as a RAM DISK) 128Kbytes min. (ROM-based; but can be used with MSX2 only)
- **Drive Assignments**
  1. Floppy-Disk Based CP/M Software:
    - A: FDD (1st drive)
    - B: FDD (2nd drive)
    - C: HDD (1st 5Mbyte area)
    - D: HDD (2nd 5Mbyte area)
    - E: HDD (3rd 5Mbyte area)
    - F: HDD (last 5Mbyte area)
    - G: FDD (1st drive, with 1DD diskette)
    - H: RAM DISK
  2. ROM-Based CP/M Software:
    - H: RAM DISK

CP/M ver. 2.2™ is a Resistered Trademark of Digital Research Inc.

## NOTE:

Drive G is used for read/write with the 1DD diskette.  
 Drive B can be accessed with a double-drive system. Access is not possible with a single-drive system.  
 Single-drive or double-drive specification is done at the time of system generation.  
 Drive B is 2DD when the system disk (Drive A) is driven with 2DD, and becomes 1DD when Drive A is 1DD.  
 The reason for the 1DD correspondence of Drive A and Drive B is so that CP/M software can be used with a special-purpose 1DD floppy disk unit. It should generally be used with a 2DD system.  
 When using floppy-disk based CP/M software, make sure the system disk is inserted in Drive A. When the system is reset, Drive A is accessed. If no disk is inserted, an error status will be issued.  
 For details regarding hard disks, please refer to the manual supplied with the hard disk unit.  
 For ROM-based CP/M software, when software is driven, system software and utility software are sent from the ROM to the RAM DISK, and are booted from the RAM DISK.

- **System Generation**

With a floppy-disk based system, a 1DD or 2DD system should be designated at the time of system generation. Make sure network access is terminated before performing system generation.

- **Utility Software**

To run a program, input the program filename, leaving off the '.COM'.

## 1. Disk-Based CP/M Software:

SYSGEN.COM	NUTL.COM
FORMAT.COM	NPIP.COM
MOVCPM.COM	TIME.COM
PIP.COM	BASIC.COM
SUBMIT.COM	SETLIMIT.COM
XSUB.COM	
ED.COM	
ASM.COM	
DDT.COM	
LOAD.COM	
STAT.COM	
DUMP.COM	
DUMP.ASM	

## NOTE:

The usual CP/M system utility software is from SYSGEN.COM to DUMP.COM. For details, refer to the explanation which appears later.

Since SYSGEN.COM and FORMAT.COM are customised for MSX computers, please refer to next page for the copying procedure.

From NUTL.COM to SETLIMIT.COM are Yamaha original utility software. They are explained below.

<b>NUTL.COM</b>	This is utility software for the network. Please refer to the Classroom Network manual.
<b>NPIP.COM</b>	This is utility software for the network. Please refer to the Classroom Network manual.
<b>TIME.COM</b>	This software displays the clock. When the software is run, the data is displayed, along with the question, "OK? (y/n)". To correct the data, input "n". If the data is correct, input "y".
<b>BASIC.COM</b>	This is used to move from CP/M software to BASIC. Since the disk is accessed when moving to MSX BASIC, take out the CP/M software disk beforehand.
<b>SETLIMIT.COM</b>	This is a partition program for using a hard disk.

## 2. ROM-Based CP/M Software:

<b>SUBMIT.COM</b>	<b>XDIR.COM</b>
<b>XSUB.COM</b>	<b>NUTL.COM</b>
	<b>NPIP.COM</b>
	<b>BASIC.COM</b>

### NOTE:

SUBMIT.COM and XSUB.COM are the usual CP/M system utility software. For details, refer to the explanation which appears later.

From XDIR.COM to BASIC.COM are Yamaha original utility software. They are explained below.

<b>XDIR.COM</b>	This is the directory software. It displays the usage volume of each file, and the unused volume of the RAM disk.
<b>NUTL.COM</b>	This is utility software for the network. Please refer to the Classroom Network manual.
<b>NPIP.COM</b>	This is utility software for the network. Please refer to the Classroom Network manual.
<b>BASIC.COM</b>	This is used when moving from CP/M software to MSX BASIC.

- **Regarding Handling of Floppy Disks**

Make sure you use disks that are copies of the originals.

- **Copying Procedure**

### 1. Format:

First, insert the CP/M software system disk in Drive A, and press the reset button to drive the CP/M software.

A Prompt: When the A > prompt is displayed, input "FORMAT" and press the return key.

The question "Drive name? (A,B)" will be displayed. For the time being, input "A" here. By doing this, formatting occurs in Drive A.

Next, the following will be displayed:

- 1 - Double Sided
- 2 - Single Sided

If using a double-drive disk, input "1"; for a single-drive disk, enter "2". The message "Strike a key when ready" is displayed. Remove the system disk and input the disk to be formatted. The formatting procedure starts when you press a key. The message "Format complete" is displayed when the process is finished.

NOTE:

Avoid formatting with Drive B as much as possible.  
When formatting with Drive B, after formatting, be sure to press the Reset switch with the CP/M system disk inserted in Drive A to perform system generation once again. If system generation is performed without first resetting the system, an error will be displayed.

2. System Generation:

Next, CP/M system software is copied to a formatted disk.

First, remove the formatted disk, and insert the system disk in Drive A. After inputting "SYSGEN", press the return key.

The following is displayed:

```
MSX CP/M SYSGEN '87 JAN
Insert system disk on A:
Are You Ready? (Y/N)
```

Input "Y". The disk begins to rotate, and the following is displayed:

```
Reading system track
Disk is 2DD (or 1DD)
How many drives? (1/2)
```

Input the number of MSX drives.  
The following will be displayed:

Insert destination disk on A:  
then push any key

Take out the system disk and insert the disk you want to copy. Press any key. When the following is displayed, the procedure is finished.

```
Disk is 2DD (or 1DD)
System track copy complete!!
```

3. Software Copying of System Disk:

Replace the disk with the system disk.

For double-drive systems:

Insert the original system disk in Drive A and the disk to be copied (after system generation) in Drive B. Input "PIP B: = A: \*. \*" and press the return key. The name of the copied file is displayed. When A > is displayed, the procedure is finished.

For single-drive systems:

When copying to a 2DD disk, copy from a 1DD disk to the 2DD disk (Drive A) after first copying to the 1DD disk (Drive G). A 2DD disk can be used to format a 1DD disk.

Input "PIP G: = A: \*. \*" and press the return key.

The filename read from Drive A is displayed, along with the message.

Change Disk for G: then Strike a Key

Substitute the 1DD disk and press any key. The following message is displayed:

Change Disk for A: then Strike a Key

Substitute the 2DD disk and press any key. Repeat this procedure until A > is displayed.





Next, copying from a 1DD disk to a 2DD disk is explained.

Insert the 1DD disk and input "PIP A: = G: \*. \*". Then press the return key. Continue with the same procedure used for copying from a 2DD disk to a 1DD disk.

- **When Using Russian**

CP/M software uses a 7-bit code system. Russian can be used only after utility software and application software for that purpose are run. (However, it cannot be used if the software ignores the 8-bit code.)

- **The following keys cannot be used on the MSX keyboard, for the CP/M system:**

<b>ESC</b>	key	
<b>Cursor</b>	key	   
<b>SELECT</b>	key	
<b>STOP</b>	key	
<b>CLS</b>	key	
<b>INS</b>	key	
<b>DEL</b>	key	
<b>GRAPH</b>	key	
<b>PYC</b>	key	Available for application software supporting Russian, such as NUTL.COM.

Refer to the application manual of each software, on the availability of the above keys.

- **Function key assignment**

As in BASIC, 10 popular commands are assigned to keys F1 through F10. The commands that are assigned to F1 through F5 are normally displayed on the screen. Press the SHIFT key to display the rest, F6 through F10.

## **Chapter 2**

---

# **REFERENCE GUIDE TO CP/M ver. 2.2™ COMMANDS AND PROGRAMS**

## INTRODUCTION

This is a reference guide for the CP/M ver. 2.2 operating system commands and programs. The programs and commands have been organized alphabetically to aid the user in finding them.

NOTE:

As FORMAT and SYSGEN of the Transient command are customised for MSX computers, please refer to the copying procedure in Chapter 1 for proper operations.

The following format is used throughout this manual:

In front of the title of the program or command is an identifier that defines whether it is a transient command program (to be found on the CP/M diskette) or an built-in command (that remains resident as an integral part of MSX). If it is Transient Command, then a drive specifier must precede the program name if it is on a disk other than the current logged drive. For example, A>B:PIP A:=B:\*. \* <cr> directs the computer to find the program PIP.COM on drive B:.

The heading "Purpose" gives a brief description of the program or command. "Syntax" lists the various command line formats that can be used to execute the program. "Usage" gives a detailed description of the program or command, and the different ways (arguments and parameters) of invoking it. Where necessary, examples are given to clarify the text.

Certain variable names are used throughout the text, and are described below.

<b>DRIVE</b>	— Any legal drive letter from A to H used in CP/M software.
<b>FILENAME</b>	— Any legal filename used in CP/M software.
<b>EXT</b>	— Any legal file extension used in CP/M software.
<b>dnum</b>	— A decimal number.
<b>xnum</b>	— A hexadecimal number.
<b>'XXX'</b>	— Any file extension is indicated in parentheses. E.g. '.BAS'.
<b>&lt; cr &gt;</b>	— Enter a carriage return.
<b>^X</b>	— Hold down the control key while depressing the character to the right. E.g. ^C indicates pressing the control (CTRL) key plus the charater key 'C'.

It should be also noted that any command program file (i.e., with the extension '.COM') may be invoked in CP/M software by entering the FILENAME followed by a < cr >.



## CONTROL KEYS

Function	Key	Action
Last character erase	<b>BS</b> CTRL-H	Erases the last character input, and backs up one space.
Line erase	CTRL-U  CTRL-X	Displays “#” and moves the cursor to the next line to wait for the next command. Erases the line in which the cursor is located, and moves the cursor to the beginning position of the line (the display will also disappear.).
System re-start (Warm boot)	CTRL-C	Re-boots the CP/M. Stops all the operations in progress.
Re-start for new disk initialization (Warm boot)	CTRL-C	Enables the write operation on the newly inserted disk after disk exchange.
Printer echo and Printer interruption (toggle)	CTRL-P	Sends input data or screen messages back to the printer. On-coming CTRL-P input toggles ON/OFF for the printer echo.
Temporary screen interruption and re-start (toggle)	CTRL-S	Puts the screen on hold and releases upon receipt of the next CTRL-S input.

Refer to the Transient Command for the availability of the control characters in each transient command function.

- Purpose:** ASM.COM™ assembles 8080 assembly language code, which has been previously saved as an ASCII text file. ASM.COM generates binary code in Intel hexadecimal format, ie '.HEX' files.
- Syntax:** 1 ASM FILENAME  
2 ASM FILENAME.
- Usage:** ASM.COM generates machine language (binary code) in Intel hexadecimal notation from 8080 assembly language source code. The generated '.HEX' file can then be "loaded" with the LOAD.COM command of the CP/M software to create an executable program '.COM' file. ASM.COM also generates a listing file which contains the original assembly language source code, complete with the generated hexadecimal code, including error flags.

The source filename must have the extension '.ASM'. as follows;

The first syntax is used when the file to be assembled is on the current active drive, and the generation of both the '.HEX' and '.PRN' files is desired on the current drive.

The second syntax is used to select a different drive for either the source file or the generated files, or if the generation of either the '.HEX' or '.PRN' files is to be omitted, or if the listing file is to be redirected to the console (in this latter case the '.PRN' file will not be created).

The 3 letters after the period designate, respectively, the name of the drive in which the source file exists, the name of the drive in which the '.HEX' file is generated, and the name of the drive in which the '.PRN' file is generated. The drive letters 'A' to 'H' are legal drive specifiers. Two additional options ('X' and 'Z') are available:

- 1 'S' (SOURCE) Selects the drive containing the original source file.
- 2 'H' (HEX) Selects the drive that should receive the assembled '.HEX' file output. If 'Z' is specified here, the '.HEX' file will not be generated.
- 3 'P' (PRN) Selects the drive that should receive the print listing file '.PRN'. If 'Z' is specified here, the '.PRN' file will not be generated. If 'X' is specified, the listing file and the error messages will be redirected to the console display.

**Examples:**

**A> ASM TEST**

This assembles the file TEST.ASM found on the current active drive A: generating both the hexadecimal file (TEST.HEX) and the listing file (TEST.PRN) on the current active drive.

**A> ASM TEST.BCX**

The first option 'B' directs the assembler to find the source file TEST.ASM on drive B:. The second option 'C' causes the assembler to generate the hexadecimal file TEST.HEX on drive C:, and the third option 'X' redirects the listing file to the console display.

ASM.COM™ is a Resistered Trademark of Digital Research Inc.

**Purpose:** The Dynamic Debugging Tool (DDT™) is used to test, debug, and perform minor alterations to a machine language program.

**Syntax:** 1 DDT  
2 DDT DRIVE:FILENAME.EXT

**Usage:** The first syntax is used to load DDT.COM into the transient program area of the computer. Once loaded, it awaits further instructions from the user.

The second syntax is used to load both DDT and the user specified program file into the transient program area. DRIVE (optional) indicates the drive that contains the source file, FILENAME is any legal filename in CP/M software, and EXT is the extension (normally '.HEX' or '.COM'). An extension must be specified with the filename. The user program can then be modified, tested, or extended using the various DDT built in commands which are listed below.

**Examples:**

A> DDT <cr>

This loads DDT into the base page of the transient program area.

A> DDT B:MYFILE.COM <cr>

This loads the program MYFILE.COM located on drive B: into the base page of the transient program area and loads DDT.COM into a higher memory location. When DDT is resident in memory it displays a hyphen as a command prompt. Program control remains with DDT unless the 'G' (Execute program) command is issued.

DDT is a powerful debugging utility program that enables the programmer to fully test and debug machine language programs. It can be used to

- Load a machine language program into memory.
- Make changes to the program in memory.
- Change the parameters of disks.
- Examine and modify memory contents.
- Enter assembly language instructions.
- Install driver routines.
- Disassemble a program section (machine language to assembly language).
- Examine and modify the CPU's register contents.
- Trace the execution of a program step by step.
- Run a program in stages and examine the results of each stage.

**DDT Commands:**

Axnum <cr> (Assembly instructions)

Enter assembly instructions beginning at the hexadecimal address specified (xnum). The mnemonic and operand entered should be separated by a single space. Entry is terminated by entering a period instead of an instruction followed by <cr>, or by entering a <cr> at the beginning of a new line.

**Example:** -A0200 <cr>  
0200 MVI A,7  
0201 . <cr>

DDT™ is a Resistered Trademark of Digital Research Inc.

DDT displays a question mark if it does not recognize your instruction and redisplay the memory location.

**Dxnum1,xnum2<cr> (Display memory)**

The parameters xnum1 and xnum2, respectively, representing the starting and ending addresses, are optional. If the addresses are not specified, the memory display begins at the current memory location.

In each case the display line output consists of first the memory address, followed by 16 hexadecimal numbers representing the machine code, followed by 16 ASCII characters representing the character value of the code (if the ASCII value is not displayable, a period is displayed instead).

Normally 12 lines (192 bytes) are displayed at one time.

**Fxnum1,xnum2,xnum3<cr> (Memory fill)**

This command fills a specified block of memory with the single byte value xnum3. The block's starting address (xnum1), and ending address (xnum2) must also be specified.

**Example:**       -F0500,0800,00<cr>

Places the null value 00 into the memory block, from 0500 to 0800 inclusive.

**Gxnum1,xnum2,xnum3<cr> (Execute)**

This command executes the program in memory. A starting address xnum1 may be optionally specified. The two additional parameters, xnum2 and xnum3, are both optional and represent breakpoint addresses. If the program encounters either of these addresses while running, execution is terminated and control returns to DDT.

If the program runs to completion in any of the above cases, control returns to the CP/M operating system.

**Hxnum1,xnum2<cr> (Hexadecimal math)**

This command computes both the sum and the difference of the two hexadecimal numbers, xnum1, and xnum2. The first number displayed is the sum, the second, the difference.

**IFILENAME.EXT<cr> (Input filename)**

This command places the name of the file FILENAME.EXT into the FCB. This is used to set up the 'R'(read file) command.

**Lxnum1,xnum2<cr> (List memory)**

The L command is used to disassemble a portion of memory, i.e., produce assembly language code from machine code. If L<cr> is entered the listing begins from either 0100 hex, or the last listed address. Optionally, a starting address xnum1 and/or ending address xnum2 may be specified.

**Mxnum1,xnum2,xnum3<cr> (Move memory)**

This command moves a block of memory. xnum1 is the start of the block, xnum2 is the end of the block, and xnum3 is the starting point of the new location. This command performs a straight move, byte for byte, and does not make any recalculations of memory addresses. All three numbers must be entered after the 'M' for this command to function.

**Rxnum <cr> (Read a file)**

This command reads the file listed in the FCB block (see the 'I' command above) and loads it into the memory location starting at the address indicated by xnum. If xnum is omitted then the program is loaded at 0100 hex.

**Sxnum <cr> (Set memory)**

This command enables the user to display or change the contents of memory. DDT displays the address and the current value. Enter either a new value followed by <cr> to change the value, or only <cr> to leave the value unchanged.

**Example:**       -S0200 <cr>

0200 AF B7 <cr> Changes the memory at 0200 from AF hex to B7 hex.

0201 CD <cr>   Leaves the value CD hex in memory at 0201 hex.

Entering <cr>   twice exits the 'S' command. See the section on SAVE.COM for further details of saving an altered program.

**Tdnum <cr> (Trace execution)**

Traces program execution and displays the contents of the CPU registers, one instruction at a time. The number dnum indicates how many instructions to execute. Pressing any key during execution returns control to DDT. The trace display appears as follows:

C0Z0M0E0I0 A=01 B=05C3 D=0043 H=0120 S=0250 P=113 ADI A,05

C=carry flag Z=zero flag M=minus flag E=even parity flag I=intermediate carry flag. The current value of the flags is listed following the initial. For example C0 means that the carry flag is set at zero. The letters A,B,D,H represent the CPU registers and their current values. S represents the current stack pointer and P the current program counter. ADI A,05 is the next instruction to execute.

**Udnum <cr> (Untrace execution)**

This command is the same as the TRACE command except that it displays only a single line of information after executing dnum instructions. The line displayed is the CPU register status after completion of the final instruction.

**X <cr> (Examine the CPU)**

Displays the current status of the CPU in the same format as the TRACE and UNTRACE commands. A single register can be displayed by entering the initial (see TRACE command) of the register. For example XC <cr> would display the contents of the carry flag; XA <cr> would display the contents of the accumulator.

## DIR

Built-in Command

---

**Purpose:** Displays a directory list of disk files (in the current user area).

**Syntax:** 1 DIR <cr>  
2 DIR DRIVE:FILENAME.EXT <cr>

The wildcards '\*' and '?' may be used to search for file groupings.

**Examples:**

1 A> DIR <cr>

Lists all the files on the current drive A:.

2 A> DIR B: <cr>

Lists all the files on disk drive B:.

3 A> DIR \*.BAS

Lists all the files on the current drive A: that have the extension .BAS.

4 A> DIR TEST???.\*

Lists all files on the current drive A: that begin with the string "TEST" followed by any three characters, and any extension. For example, TEST001.BAS, TESTING.ASM, and TESTTXT.DOC would all be displayed on the directory list, but the file TESTFIVE.DOC would not be displayed as it has 4 characters after "TEST".

5 A> DIR TEST\*.\*

Lists all the files in example 4 above including TESTFIVE.DOC.

**Purpose:** DUMP displays the contents of any file to the display console in hexadecimal notation.

**Syntax:** A>DUMP DRIVE:FILENAME.EXT

DRIVE is optional and is used only if the source file is not on the current drive. FILENAME.EXT is the full file name of the source file. Wildcards are not permitted.

**Description:** The display line format starts at the left with the starting address of the current line followed by 16 hexadecimal numbers showing the value of each byte.

**Examples:** A>DUMP PROG.COM <cr>  
A>DUMP B:MYFILE.HEX <cr>

**Purpose:** ED.COM is a compact text editor. The program contains many built-in commands that let you create, display, delete, modify, and append any text file.

**Syntax:** ED FILENAME.EXT < cr >

When ED is first loaded, it creates a memory text buffer into which the user can selectively load all or part of the selected file FILENAME.EXT. ED automatically creates a temporary file using the same name as the source file, but with the extension \$\$\$\$. This temporary file is used to store the appended file when the memory buffer becomes full. ED also creates a backup file, with the extension BAK, and retains this as a copy of the original file when editing is finished. Text may be inserted into the file being edited from selected "library" files to speed up repetitive entries. A full summary of ED commands is given in appendices A and B.



**Purpose:** ERA erases a specified file or files from the current user area.

**Syntax:** ERA DRIVE:FILENAME.EXT

**Usage:** Wildcards are allowed when specifying FILENAME.EXT; however, care should be taken as it is easy to erase a file by mistake. Files that have been erased are difficult or impossible to recover.

**Examples:**

A>ERA LETTER.DOC

Erases the file LETTER.DOC residing on drive A:.

A>ERA \*.BAS

Erases all the files with the extension .BAS that reside on drive A:.

A>ERA B:TEST\*.\*

All the files on drive B: that start with the characters "TEST", for example, TEST001.DOC, TEST1.DOC, TESTDOC.001, and TESTDOC1.1 would be erased.

**Purpose:** MOVCPM is used to reconfigure CP/M to fit into a system with a different memory size.

**Syntax:** MOVCPM dnum \*

**Usage:** Dnum is a decimal number, specifying an optional memory size in kilobytes (e.g., '48' for a 48K system). The wildcard '\*' may be used in place of dnum which will cause MOVCPM to calculate and use the size of the current computer's total RAM (random access memory). Out of 64KB, a maximum of 56KB can be used as memory area (TPA). The remaining 8KB are used by the system. In either case, a new version of CP/M software will be generated to fit the specified or calculated size.

The second asterisk, following either dnum or \* is optional, but if used will cause MOVCPM to leave a copy of the newly generated system in the TPA (Transient Programmable Area). From there, the user may use either the SAVE or SYSGEN program to create a disk file of the new system. If the second asterisk is not specified, then MOVCPM will execute the new CP/M software without retaining a copy.

**Examples:**

A> MOVCPM 32<cr>

Would create a 32K version of the CP/M software, and execute it without storing it in the TPA.

A> MOVCPM \* \* <cr>

Would create a new copy of CP/M equal in size to the maximum RAM of the computer, and leave a copy of it in the TPA where it could be saved to disk with either SAVE or SYSGEN.

**Purpose:** The Peripheral Interchange Program (PIP) copies information from one source to another.

**Syntax:**

- 1 PIP <cr>
- 2 PIP DRIVE1:=DRIVE2:\*. \* <cr>
- 3 PIP DRIVE1:=DRIVE2:FILENAME.EXT[OPT] <cr>  
( DRIVE2, and OPT are optional)
- 4 PIP DRIVE1:FILENAME1.EXT = DRIVE2:FILENAME2.EXT[OPT],  
FILENAME3.EXT[OPT]... <cr>  
(DRIVE1, DRIVE2, OPT, and FILENAME3.EXT are optional)

In the above, FILENAME.EXT refers to any legal filename in CP/M software or any standard device name used in CP/M software (eg. CON:).

The wildcards '\*' and '?' are permitted as part of any source filename.

**Usage:** The first syntax is used to load PIP.COM into the program memory. Once loaded, PIP displays an asterisk and waits for further user input. This format is useful when a series of copy procedures is desired, or when it is necessary to remove the system disk in order to make copies from non-system disks. Control remains with PIP until either ^C (control 'C') or <cr> is entered at the beginning of a new line. The system disk should be reinserted before exiting PIP.

The second syntax uses wildcards (\*.\*) to copy all the files from the specified source drive to the specified destination drive.

The third syntax is used to copy a single source file to a destination drive or device. A filename may be specified for the receiving file. If no destination filename is selected, a new file is created with the same name as the source file. Option parameters may be specified following the source filename (see appendix E for a complete listing).

The fourth syntax is used when copying specified multiple files to a new single file or device. A single filename must be specified as the receiving file. The source files must be separated with a comma, and any options must be specified immediately following the filename.

**Example 1:**

```
A>PIP<cr> (Loads PIP.COM into the program memory and displays a '*')
*:NEWFILE.DOC=B:OLDFILE.DOC<cr>
(Copies B:OLDFILE.DOC to A:NEWFILE.DOC)
*B:=A:LETTER.TXT<cr> (Copies A:LETTER.TXT to B:LETTER.TXT)
**^C (Entering ^C exits PIP.COM and returns control to CP/M)
A>
```

**Example 2:**

```
A>B:PIP A:=B:*. * <cr>
Copies all the files residing on drive B: to drive A:. In this example, PIP.COM was resident
on drive B:.
```

**Example 3:**

A> PIP LST:=B:MYLETTER.DOC <cr>

Sends a copy of the file MYLETTER.DOC residing on drive B: to the standard device LST: used in the CP/M software, which is normally configured as the printer.

**Example 4:**

A> PIP ARCHIVE.DOC=MEMO001.DOC[E],MEMO002.DOC[E],MEMO003.DOC[E] <cr>

Copies three files, MEMO001.DOC, MEMO002.DOC, and MEMO003.DOC to the file ARCHIVE.DOC. If the file ARCHIVE.DOC already exists, it will be automatically erased before the copying starts. The selected option 'E' echoes the data to the console, and the option 'V' verifies the copying of the file MEMO001.DOC by comparing the newly created file to the memory buffer.

**Example 5:**

A> PIP STORAGE.TXT=RDR: <cr>

Copies the input from the standard reader device RDR: used with CP/M software to the file STORAGE.TXT. When copying from a device, a receiving filename must be specified.

**Note:** In all cases PIP creates a new file for the receiving file. If a file already exists with the same name as the receiving file, it will be erased. Care should be taken, especially when using wildcards to avoid accidentally erasing a file.

**Example 3:**

A> PIP LST:=B:MYLETTER.DOC<cr>

Sends a copy of the file MYLETTER.DOC residing on drive B: to the standard device LST: used in the CP/M software, which is normally configured as the printer.

**Example 4:**

A> PIP ARCHIVE.DOC=MEMO001.DOC[EV],MEMO002.DOC[E],MEMO003.DOC[E]<cr>

Copies three files, MEMO001.DOC, MEMO002.DOC, and MEMO003.DOC to the file ARCHIVE.DOC. If the file ARCHIVE.DOC already exists, it will be automatically erased before the copying starts. The selected option 'E' echoes the data to the console, and the option 'V' verifies the copying of the file MEMO001.DOC by comparing the newly created file to the memory buffer.

**Example 5:**

A> PIP STORAGE.TXT=RDR:<cr>

Copies the input from the standard reader device RDR: used with CP/M software to the file STORAGE.TXT. When copying from a device, a receiving filename must be specified.

**Note:** In all cases PIP creates a new file for the receiving file. If a file already exists with the same name as the receiving file, it will be erased. Care should be taken, especially when using wildcards to avoid accidentally erasing a file.

## REN

Built-in Command

---

**Purpose:** Renames a file on the same disk drive.

**Syntax:** REN DRIVE:NEWFILE.EXT = DRIVE:OLDFILE.EXT

This command is used to rename a file. The wildcards '\*' and '?' are not permitted as part of the filenames. DRIVE is an optional specifier and can be used for either one or both files; however, if DRIVE is specified for both files, it must be the same for each. If DRIVE is specified for only one file, then it automatically applies to both.

**Example:**

A> REN B:FINAL.DOC = SAMPLE.DOC

Would rename the file residing on drive B: called SAMPLE.DOC, as FINAL.DOC. FINAL.DOC would reside on drive B:.

**Purpose:** SAVE writes a copy of the TPA starting from 0100 hex to a disk file.

**Format:** SAVE dnum FILENAME.EXT

**Usage:** SAVE is used to write a portion of the TPA to a disk file. The amount of memory to save must be user specified by dnum, which is a decimal number representing the number of 256 byte sized "pages" to save. One of the more common uses for the SAVE command is to make a copy of a file that has been loaded by DDT.COM. When a file is loaded into memory by DDT, the initial information displayed helps determine the number of pages in a file.

**Example:** A> DDT MYPROG.HEX  
DDT VER 2.2  
NEXT PC  
1450 0100  
-G <cr> (Executes the program and returns to the CP/M software)

A> SAVE 20 MYPROG.COM <cr>

The number of pages was selected as 20 by ignoring the last 2 digits '50' listed under NEXT, and converting the first 2 digits '14' from hex to decimal. The only exception to this rule is, if the last 2 digits are 00, then 1 should be subtracted from the final decimal number after the conversion.

**Purpose:** STAT displays and/or sets the status of files, disk drives, or drivers.

**Syntax:** File and disk status

1 STAT <cr>

Displays the disk attribute and the amount of free space for all drives.

2 STAT DRIVE: <cr>

The same as 1 except for a specified drive.

3 STAT DRIVE:FILENAME.EXT <cr>

Displays the status of single or multiple files (using wildcards).

4 STAT DRIVE:FILENAME.EXT \$ATR <cr>

Changes the attribute of single or multiple files.

Disk status

5 STAT DRIVE:DSK: <cr>

Displays the current disk configuration.

6 STAT DRIVE: =R/O

Temporarily changes the disk attribute to R/O.

Device status

7 STAT DEV: <cr>

Displays the current device assignments.

8 STAT VAL: <cr>

Displays the logical devices and their possible physical assignments. Also displays a brief summary of the STAT command line.

9 STAT LOGICAL: =PHYSICAL: <cr>

Assigns the specified physical device to the specified logical device.

10 STAT USR: <cr>

Displays the current user number and lists all the user numbers which contain active files.

Possible file attributes: (\$ATR)

R/O Read only file            R/W Read and Write file

SYS System file            DIR Directory file (default attribute)

Refer to appendix C for a description of logical and physical drives.

The first three syntaxes are used to display information about a file or files on any disk drive. The fourth is used to change file attributes.



**Example 1:**

A> STAT A:\*. \* <cr>

RECS	BYTES	EX	ACC	
4	3K	1	R/W	A:MYMEMO.DOC
3	2K	1	R/O	A:SPECIAL.DOC
56	8K	1	R/W	A:(XDIR.COM)

BYTES REMAINING ON A: 238K

In the above example, the following headers are used by STAT:

- RECS — The number of 128 byte records used by the file.
- BYTES — The length of the file in 1K blocks. 1K equals 1024 bytes.
- EX — The number of extents used by the file.
- ACC — The file attribute for access, R/O (read only) or R/W (read and write).

The parentheses enclosing XDIR.COM indicate that it is a system (\$SYS) file. This attribute can be set as in the following example 2 (note that the character '\$' must precede the attribute).

A> STAT XDIR.COM \$SYS <cr>

This would set the attribute of the file XDIR.COM to SYS, making it a system file. A system file is not listed when using the DIR command.

The fifth and sixth syntaxes are used to display information about a particular disk drive, and its current data storage configuration.

**Example 3:**

A> STAT B:DSK: <cr>

B: DRIVE CHARACTERISTICS  
 2432: 128 BYTE RECORD CAPACITY  
 304: KILOBYTE DRIVE CAPACITY  
 128: 32 BYTE DIRECTORY ENTRIES  
 128: CHECKED DIRECTORY ENTRIES  
 256: RECORDS/EXTENT  
 32: RECORDS/BLOCK  
 64: SECTORS/TRACK  
 2: RESERVED TRACKS

A>

The most pertinent information in the above display is, "KILOBYTE DRIVE CAPACITY," which indicates the total amount of space available on an empty disk, and "32 BYTE DIRECTORY ENTRIES," which is the maximum number of allowable individual file names per disk .

If the disk identifier (A:) is omitted, the characteristics for all drives are displayed.

The syntaxes from seven through ten are used for displaying and setting the characteristics of the device drivers used with CP/M software.

**Example 4:**

A> STAT CON: = CRT:, LST: = LPT: <cr>

This sets the logical devices CON: and LST: to the respective physical devices CRT: and LPT:. A full list of the devices used with CP/M software is given in appendix C.

**Example 5:**     A> STAT DEV: <cr>

CON: IS CRT:

RDR: IS UR1:

PUN: IS UP2:

LST: IS LPT:

A>

This displays the logical devices in the left-hand column and their respective current physical device assignments.

For a summary of which physical devices may be assigned to the individual logical devices, enter STAT VAL: <cr> .

**Purpose:** SUBMIT automates command line entry by allowing the user to indicate a file with the extension .SUB consisting of a series of command line instructions which the computer will then execute in series.

**Syntax:** SUBMIT FILENAME <cr>  
SUBMIT FILENAME var1 var2 var3... <cr>

**Usage:** The first syntax is used when a .SUB file containing a series of commands is used that does not need any variable information for any of the command lines. For example, if the file MYSTAT.SUB has been created with the following lines:

```
DIR A:*. * <cr>
STAT A:*. * <cr>
```

then typing the following command line:

```
A> SUBMIT MYSTAT <cr>
```

would first invoke the DIR program to list the current directory of drive A:, and then invoke STAT.COM to show the current disk status.

The second syntax is used when variables are required in the control line. For example, if the file EDIT.SUB contained the lines:

```
ED $1.$2 <cr>
PIP B: = A:$1.$2 <cr>
STAT B:*. $2 <cr>
```

then typing the command:

```
A> SUBMIT EDIT MYFILE DOC <cr>
```

would cause the three commands in the file EDIT.SUB to be executed substituting "MYFILE" for \$1 and ".DOC" for \$2 every time they were encountered. In effect, the above would execute ED.COM with the working file MYFILE.DOC, upon exiting the ED program, PIP would be invoked to make a copy of MYFILE.DOC on drive B:, and finally, STAT would be invoked to display the statistics of all the files on drive B: with the extension DOC.

## TYPE

Built-in Command

---

- Purpose:** TYPE displays the contents of an ASCII file on the console display.
- Syntax:** TYPE FILENAME.EXT <cr>
- Usage:** If a non ASCII file is specified as the FILENAME.EXT (e.g., PIP.COM) then the output to the screen will be garbled and will not be meaningful (see DUMP.COM for displaying non ASCII files). If a printer is connected, a hard copy can be generated with TYPE by entering ^P before typing the command line.
- Example:** A> TYPE MYLETTER.DOC <cr>

**Purpose:** XSUB provides a way to automatically input information into the command line in response to a program's questions. XSUB.COM is a subset of SUBMIT.COM and can only be invoked from within a SUB file. For example, if the SUB file ERASE.SUB is created with the following lines:

```
XSUB <cr>
ERA *.* <cr>
Y <cr>
```

then the command line A > SUBMIT ERASE would automatically issue the following lines:

```
A > XSUB <cr>
(XSUB ACTIVE)   (CP/M message)
A > ERA *.* <cr>
ALL (Y/N)? Y <cr> (The ERA prompt Y/N is answered 'Y' by the .SUB file.)
A >
```

Using a combination of SUBMIT and XSUB can in some instances save time by automating console input and output. If, however, a program requires variable input at run time, XSUB will not be suitable.



# APPENDIX

---

## APPENDIX A

### ED.COM CONTROL CHARACTERS

KEY	FUNCTION
<b>^C</b>	Resets the system, exits ED.COM and returns to CP/M software.
<b>^E</b>	Moves the cursor to the next command line to continue a long command string without executing it.
<b>^H</b>	Erases the last character typed and repositions the cursor.
<b>^I</b>	Performs a horizontal tab, moving the cursor 7 columns.
<b>^J</b>	Performs a RETURN.
<b>^M</b>	Performs a RETURN.
<b>^L</b>	Used during search and substitute commands as a replacement for a carriage return generated by a RETURN in a string.
<b>^R</b>	Retypes the current line.
<b>^U</b>	Deletes the current line.
<b>^X</b>	Erases the current line and repositions the cursor at the start of the line.
<b>&lt;cr&gt;</b>	Executes the current command line, or starts a new text line.
<b>DEL</b>	Deletes the character to the left of the cursor.
<b>^Z</b>	Terminates the insertion 'I' command, or separates text strings in search and substitutions, or uses it as an end of file marker.
<b>^P</b>	Toggles the printer screen echo On/Off.
<b>^S</b>	Pauses the display of a long file. Striking any key continues the display.
<b>BREAK</b>	Stops execution of the current command.



## APPENDIX B

### ED.COM COMMAND SUMMARY

Note: In this appendix, the characters CP refer to the character pointer.

Command	Meaning
<b>#A</b>	Appends or copies a number of lines from the source file into the buffer. 'A' alone moves a single line. #A copies the entire file (or until the buffer is full). 0A copies the source file until the buffer is at least half full.
<b>dnumW</b>	Writes "dnum" lines from the edit buffer to the temporary file. 'W' alone copies a single line. #W copies the entire buffer, and 0W copies the buffer until it is half empty.
<b>E</b>	Ends the editing session. All text in the edit buffer is copied to the temporary file. All the text remaining in the source file is appended to the temporary file. The source file is renamed as the BAK file. The temporary file is renamed as the source file. The library file is erased, and control returns to the CP/M software.
<b>H</b>	Ends the editing session as with the E command, but control returns to ED.COM with the new source file loaded.
<b>O</b>	Erases the edited file, flushes the buffer, erases the temporary file, and returns to the start of the original file.
<b>Q</b>	Quits ED.COM and returns to the system. The source file remains intact (not appended), but the temporary and backup files are erased. All work in the edit buffer is lost.
<b>R</b>	Reads the file X\$\$\$\$\$\$\$.LIB created by the X command into the edit buffer immediately following the current CP.
<b>RFILENAME &lt;cr&gt;</b>	Reads the .LIB file specified into the edit buffer immediately following the current CP. The library file is not affected.
<b>dnumX</b>	Writes dnum lines from the editor buffer to the library file X\$\$\$\$\$\$\$.LIB starting from the CP. The lines read are not deleted from the edit buffer. Refer to the R command for reading the library file.
<b>- / + B</b>	Moves the CP to the beginning '+' or to the end '-' of the edit buffer. 'B' alone is the same as '+ B'.
<b>- / + dnumC</b>	Moves the CP forward '+' or backward '-' dnum of characters. Note that a carriage return counts as 2 characters.
<b>- / + dnumD</b>	Deletes dnum of characters before '-' or after '+' the CP.
<b>I &lt;cr&gt;</b>	Insert mode. Start inserting text into the edit buffer after the CP. If an upper case I is used to initiate this mode, all text will be converted to upper case. If a lower case i is used, then text will be recorded as entered. Entering ^Z terminates the insert mode. Refer to appendix A for a summary of control commands.
<b>Istring ^Z</b>	Inserts the character sequence "string" into the edit buffer after the CP. The CP moves to the end of the new text entered.
<b>Istring &lt;cr&gt;</b>	Inserts a line of characters into the edit buffer following the CP and adds a carriage return/line feed to the end of the line.

## APPENDIX B

<b>- / + dnumK</b>	Deletes dnum lines before ' - ' or after ' + ' the CP, including the characters at the respective side of the CP.
<b>- / + dnumL</b>	Moves the CP to the beginning of the current line, and dnum lines backward ' - ' or forward ' + '.
<b>- / + dnumP</b>	Displays dnum pages (24 lines) of the edit buffer. " - dnumP" moves backward through the buffer and " + dnumP" forward. The CP is first moved to the top of the page and then the text following the pointer is displayed. "0P" displays the current page without moving the CP.
<b>- / + dnumT</b>	Displays lines of text relative to the current cursor position. " - 3T" would display the 3 lines before the CP, and " + 3T" would display 3 lines following it. If the CP is at the beginning of a line, 'T' will display the line. If the CP is within a line, "0T" will display the part of the line before it, and "0TT" will display the entire line. The single command sequence "B#T" would display the entire buffer.
<b>- / + U</b>	Translates all lower case entries into upper case. " + U" begins the translation, " - U" terminates the translation mode.
<b>- / + V</b>	Toggles the line number display on " + V" or off " - V".
<b>0V</b>	Displays the amount of free space left in the buffer and the total size of the current buffer.
<b>dnum:</b>	Moves the CP to the beginning of the specified line number dnum.
<b>- / + dnum</b>	An abbreviated form of the "nLT" command sequence, moves the CP forward or backward dnum lines, and displays the new current line.
<b>dnumFstring^Z</b>	Finds character string "string" dnum of times after the current CP, and repositions the cursor at the end of the final (dnum) match. If the search fails, the CP does not move.
<b>dnumNstring^Z</b>	Same as the F command above except that it also searches the source file for the given "string". If necessary, the edit buffer and temporary file are updated as the source is searched.
<b>dnumSfind^Zreplace^Z</b>	Searches for the string "find" and substitutes the string "replace" a total of "dnum" times. The search begins at the CP and ends at the last character in the edit buffer.
<b>dnumJfind^Zinsert^Zend^Z</b>	Searches for the string "find" and inserts the string "insert" immediately after it, and then deletes all the successive characters following "insert" until a match for the string "end" is found. The cursor is repositioned to the final character of the string "end".

## APPENDIX C LOGICAL AND PHYSICAL DEVICES FOR CP/M SOFTWARE

### Logical devices:

- CON:** Input from the keyboard, output to the display
- RDR:** Paper tape or card reader (input only)
- PUN:** Paper tape or card punch (output only)
- LST:** Listing device such as a printer (output only)

### Physical devices:

- TTY:** Slow console display (teletypewriter)
- CRT:** Fast console display (cathode ray tube)
- PTR:** Paper tape reader
- PTP:** Paper tape punch
- LPT:** Line printer
- BAT:** Batch processor
- UR1:** User defined reader 1
- UR2:** User defined reader 2
- UP1:** User defined punch 1
- UP2:** User defined punch 2
- UC1:** User defined console
- UL1:** User defined list device

**NOTE:** In the above device names, the colon is required as part of the name.

## APPENDIX D PIP.COM KEYWORDS

KEYWORD	DESCRIPTION
<b>NULL:</b>	Sends 40 null (00 hex) bytes to a device, originally intended for use with a punch device to give a blank header or trailer. For example: A> PIP PUN: = MYPROG.HEX,NULL: <cr>
<b>PRN:</b>	Sends the data to the LST: device with the following format. Tabs are expanded to eight spaces, all lines are numbered, and a form feed is sent after every 60 lines. A> PIP PRN: = MYPROG.ASM <cr>
<b>EOF:</b>	Explicitly sends an end-of-file (^Z) marker to the destination device. PIP automatically sends an EOF marker, but in certain circumstances EOF: can be used to terminate the transfer. This is especially true when transmitting between two computers. A> PIP PUN: = MYPROG.HEX,EOF:
<b>INP: and OUT:</b>	Special device drivers. Both of these devices respectively input and output information to user created destinations. The hexadecimal locations 0103, 0104, and 0105 are reserved for jump instructions to the user created input routine, and 0109 hex is where PIP looks for the input character. Locations 0106, 0107, and 0108 are reserved for jump instructions to the output routine. PIP calls 0106 after first loading register C with the character to transmit. The area from 010A hex to 01FF hex is available for the user routine.

## APPENDIX E

### PIP.COM PARAMETERS

<b>B</b>	Block transfer mode. PIP transfers information to a buffer until the ASCII character ^S is received. Once received, PIP sends the contents of the buffer to the specified destination drive. This is useful when reading data from a continuously reading device such as a paper tape.
<b>Ddnum</b>	Deletes all characters after the dnum column. PIP uses the carriage return as an index to start the column count. Useful for sending information to a device with a narrow screen.
<b>E</b>	Echoes to console display. The information being sent to a device or file is echoed to the console display while it is being copied.
<b>F</b>	Filter form feeds. PIP will remove the form feed character (0C hex) from the data stream before it is sent to the destination device or file.
<b>H</b>	Checks the data being transferred to insure that it is in proper Intel hexadecimal format. Useful for verifying the transfer of '.HEX' files.
<b>I</b>	Ignores any null records during the transfer of Intel hexadecimal format files. This option automatically sets the H option.
<b>L</b>	Translates all characters sent to lower case.
<b>N</b>	Adds line numbers followed by a colon to each line of data as it is sent. The number 2 following the parameter e.g. "N2" adds leading zeros to the line number, followed by a tab.
<b>O</b>	Transfers an object file. This parameter tells PIP that the file being transferred is a non ASCII file, and changes the method that it uses when handling the end-of-file marker. PIP automatically sets this option for files with the extension COM.
<b>Pdnum</b>	Issues a form feed after every dnum lines of data. If dnum is omitted, a default value of 60 is assumed. This option is normally used in conjunction with a printer.
<b>Qstring^Z</b>	Copies a portion of a file or device until a match for the specified string is found. The string must be terminated with a ^Z.
<b>R</b>	Allows PIP to copy a system file. The system file attribute (\$sys) is also copied to the destination.
<b>Sstring^Z</b>	Starts copying from a file or device after a match for the specified string has been found. This parameter may be used in conjunction with the Q option.
<b>Tdnum</b>	Sets tab width to dnum columns and expands tabs (^I) in ASCII files. The default number is 8 if dnum is not specified.
<b>U</b>	Translates all lower-case characters to upper-case characters during ASCII file transfer.
<b>V</b>	Verifies the data being sent to a disk file by re-reading the destination device after sending and comparing it to the information in the data memory buffer. The V must follow the first source name.
<b>W</b>	Allows PIP to write to a read only file. PIP deletes the original destination file before copying to it.
<b>Z</b>	Sets the eighth (parity) bit of ASCII characters to zero before sending them to the destination file or device. This filter is sometimes useful when sending files

## APPENDIX E

created by some word editors that use the eighth bit to indicate “soft spaces” or “soft carriage returns” which can cause problems when listing a file.

The above parameters may be used alone, or as a series of commands. For example the command line:

```
A> PIP B: = A:MYFILE.DOC[EZLQfinish^ZV] <cr>
```

would copy the file on drive A: named MYFILE.DOC to the file on drive B: named MYFILE.DOC, echo the data to the console display, change the eighth bit of each character to null, translate all the characters to lower case, stop copying after the first occurrence of the string “finish”, and verify all data received.

## APPENDIX F

### COMMON EXTENSION TYPES USED WITH CP/M SOFTWARE

Extension	Type	Example
<b>COM</b>	Necessary for any executable program file.	STAT.COM ASM.COM
<b>ASM</b>	Necessary for any assembly language text file used with ASM.COM.	MYPROG.ASM XDIR.ASM
<b>PRN</b>	This is used by ASM.COM as the listing file extension.	MYPROG.PRN XDIR.PRN
<b>HEX</b>	This is used by ASM.COM as the hexadecimal machine code file.	MYPROG.HEX XDIR.HEX
<b>BAS</b>	This indicates a BASIC source file.	SAMPLE.BAS TEST.BAS
<b>INT</b>	A compiled BASIC file.	SAMPLE.INT
<b>BAK</b>	Used by ED.COM to hold the back-up copy of the file being edited.	LETTER.BAK
<b>\$\$\$</b>	A temporary file that is normally erased automatically by the program.	MYDOC. \$\$\$
<b>SUB</b>	A text file containing a list of commands to be executed using SUBMIT.COM.	BACKUP.SUB

# YAMAHA

YAMAHA CORPORATION  
88 02 CR Printed in Japan



Этот документ подготовлен группой RBSC при участии:  
• поиск, приобретение оригинала, сканирование: Александр Страйстар  
• обработка и сборка в PDF: Алексей Сысоев  
Февраль 2023  
Сайт группы: <https://rbsc.su/>