

ISSUE 1
JUNE 15, 1975

computer systems COMMUNICATOR

HP 3000
MINI DATACENTER

2000/ACCESS REMOTE DATA
ENTRY/DATA COMMUNICATION

9600 MEASUREMENT
AND CONTROL SYSTEM

2000/ACCESS REMOTE DATA
ENTRY/DATA COMMUNICATION

editor's note

Welcome to a new world of information — the Hewlett-Packard **Computer Systems Communicator**. The **Communicator** contains the kind of information you can put to use immediately on your HP 3000, 2000, or 9600 Computer System. The information is supplied by HP Data Systems support personnel, and is designed especially for you, the Customer.

Typical features you will find in the **Communicator** are programming tips, hardware techniques, the latest software changes, and a variety of reference material — including manual availability, training schedules, new software products, and ordering information.

In addition, each issue will have a central feature or theme of special interest. This issue's theme is *Users Groups*. We have tried to introduce to you some of the user-oriented organizations available to HP Data Systems Customers. Information is provided as to the services rendered by each group and the advantages of becoming a member.

The **Communicator** is divided into several parts. The first part with its light blue border is for HP 2000 Computer Series users. The final section is bordered in dark blue and contains data relative to HP 3000 Computer products. General information and feature articles are gathered into a common center section.

contents

ABOUT THE HP 2000

Software Tips

Writing DOS-III Directives into Files	2
What the :ST,X Directive Does and Does Not Do	2
Implementing a Self-Written IBL Loader	2
Determining Program Length in RTE-B	3
Sharing I/O Slots on a 21MX	3
How to Spool in TCS	4
Reconfiguring BCS for New Interface	5

Bulletins

CWF Support Plan Modification	5
-------------------------------------	---

Documentation

Disc Operating System Manuals	6
Software Input/Output System Manuals	6
Basic Control System Manuals	6
Timeshared System Manuals	7
Language Manuals	7
Additional Manuals	7

Training Schedule

Maintenance Courses	8
Software Courses	8

Scheduled for future issues is a third section devoted to the 9600 Computer System. RTE customers should note that although not included in this issue, RTE systems will be emphasized in the next issue of the **Communicator**.

We believe you're going to find the **Communicator** an asset to your computer operation. We hope you let us know how you like our new publication. We hope, also, that you'll share your ideas with us in the form of articles for future issues or even comments to the editor. Address your correspondence to:

Editor
Computer Systems Communicator
HP Data Systems Division
Systems Engineering Department
11000 Wolfe Road
Cupertino, Ca. 95014

SUBSCRIPTION INFORMATION

The first three Preview Issues of the **Communicator** are being distributed without charge to a large list of potential subscribers. The subject of these first issues will vary considerably as we provide broad coverage of customer-oriented technical information across the entire Data Systems Division product line. The **Communicator** replaces two previous HP publications: *Computer News* and *System Software Notes*. We are sure you will judge from the Preview Issue that the **Communicator** represents a valuable service which you will want to secure for your installation. The price of this annual subscription service and the method by which the **Communicator** is to be ordered will be described in each of the subsequent issues.

FEATURING — USERS GROUPS

The HP Educational Users Group	9
HP Clearinghouse	10
The HP 3000 Users Group	11

ABOUT THE HP 3000

Software Tips

Closing Files in FORTRAN	12
Calculating the Approximate Size of Output Spoolfiles	12
Code Segment Allocation	12
COBOL Callable Intrinsic	13
Tips on Using the BASIC/3000 Compiler	15

Bulletins

Scheduled for Obsolescence	16
HP 3000 MPE Internals Course	16

Software Updates

MPE 32000C.00.05	17
HP 32212A FCOPY/3000	26
HP 32104A RPG/3000	26
HP 32213 COBOL-A/B	26
HP 32215A IMAGE/3000	27

Documentation

MPE/3000 Manuals	28
Language Manuals	29
Additional Manuals	29

Training Schedule

Software Courses	30
------------------------	----

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

This section of the Communicator contains timely how-to-do-it information contributed by HP customers and personnel. The information presented is pertinent to 21MX based systems, i.e., BCS, DOS, and Timeshared systems.

You may find just the solution you've been looking for in this section, or merely a better way to accomplish a particular task. If you've found a solution on your own, why not send it to the Communicator so that we can share it with everyone.

DOS-III

DOS-III allows directives to be stored to a file using the C option in the :STORE,S command. The logical unit that the source images are taken from cannot be the currently executing batch device. Thus if you want to store source, including directives, from cards you cannot use the card reader as the batch device.

To circumvent this restriction, merely assign an unused logical unit to the card reader and issue a batch directive to the "unused" device as in the example below.

Example	Notes
:BA,7	Batch from card reader
:ST,S,RKS,7,C	Store from card reader
ILLEGAL LUN	Error from system
JOB ABORTED!	Control returned to console
:TY	
@	
:LU,5,7	Change of Logical units
@	
:BA,5	Batch from LU 5
:ST,S,RKS,7,C	Store directive
0005 LINES	Store accomplished
:TY	Control returned to console

R. K. Strand
HP Cupertino

The :ST,X directive in DOS-III allows you to store an Assembly Language program that was assembled with an A in its ASMB control statement. The :ST,X directive does not allow you to store a "core image" of the following:

- Assembly program with an R in its control statement
- ALGOL programs
- FORTRAN programs

For instance, this directive can be used to duplicate an absolute program (such as DSGEN) from a disc file to magnetic tape or paper tape.



Paul McGillicuddy
HP Cupertino

IBL

IBL

The procedure for implementing your own Initial Binary Loader (IBL) on a 21MX includes the following steps:

1. Write the loader in absolute HP Assembler Language.
2. Assemble the loader obtaining an object tape.
3. Produce a PROM writer mask tape from the assembler object tape.
4. Burn the loader into a 1K chip using the PROM writer control program, PROM writer, and mask tape from Step 3.
5. Install the chip from Step 4 in the CPU.
6. The loader is ready to be run.

Step One

Write the loader in absolute assembler code, originating the loader at X7700, where X = 0, 1, 2, 3, 5, 7 as the memory size is correspondingly 4K, 8K, 12K, 16K, 24K, 32K.

Use 64 words or less.

All select codes in loader I/O instructions will be configured automatically at IBL time. Switch register bits 11-6 will be taken as the configuring select code. 10 (octal) will be subtracted from the configuring select code and the result added to the select code part of all loader I/O instructions, except: if the select code will not be modified. Loader constants having bit 15 on, bits 14-12 off, bit 10 on, will be interpreted as I/O instructions and will be configured as per above.

At IBL time, word 64 of the loader will be forced to the starting address of the loader in two's complement form. Word 63 of the loader will be unconditionally configured as described above for the switch register configuring select code.

If the loader device is a single channel device, use 10 (octal) as the select code portion of all loader device I/O instructions. If the loader device is a two channel device, use 10 and 11 (octal) as the select codes.

It is strongly suggested that the loader be checked out before Steps Two through Six are performed. This can be done by writing a short program to load the last 64 words of memory with the loader, which may then be executed and checked out.

Step Two

Assemble the loader obtaining an object tape.

Step Three

Run the assembler-object-to-PROM-mask tape conversion program. This program is being added to the HP Contributed Library. (If you have an immediate need for the program, write a note to the **Communicator** Editor.)

The output of the converter program is a standard PROM writer mask tape.

Step Four

Burn the loader into a 1K chip (HP #1816-0015 or 1816-0129) by running the PROM writer control program with the mask tape from Step Three as input. The PROM writer control program for BCS is 24287-60001. The PROM writer control programs for DOS are 24287-60001 and 24287-60002. Operating procedures are provided in the PROM writer manual (12909-90009), pages 2-1 through 2-6 and 3-1 through 3-5.

Step Five

Install the chip burned in Step Four into the CPU. The 21MX installation and service manual (02108-90006) describes this process on page 2-5.

Step Six

The loader may now be brought into memory by setting the S register and pressing the IBL button. Refer to the 21MX operators manual (02108-90004) page 7 for instructions.

*Jack Howard
HP Cupertino*

SYMTF = BASE + 16₈
FCORE = BASE + 15₈

Also:

<SYMTF> = contents of memory location SYMTF
<FCORE> = contents of memory location FCORE

After a program has been loaded and the RUN command has been issued, <SYMTF> and <FCORE> are modified by the system so that the difference <SYMTF> — <FCORE> is the remaining memory (in octal) for your programs.

If you wish to know the length of the program currently residing in memory, the procedure to use is as follows:

1. LOAD the program.
2. RUN it.
3. Examine the locations SYMTF and FCORE; record their contents <SYMTF> and <FCORE>; take the difference and call it PROG.
4. DELete the program.
5. Write the following bare-bones program: 100 REM
6. RUN it.
7. Repeat step 3, but call the difference NOPROG.
8. The number of memory cells occupied by the original program is PROG-NOPROG, except for the trivial difference corresponding to the bare-bones program. The amount of memory available to you with no program in the system is NOPROG.

Remember, it is necessary that the program in question be RUN before the difference <SYMTF> — <FCORE> is meaningful.

*Joe Diesel
HP Cupertino*

DETERMINING THE MEMORY REQUIREMENTS OF A 21MX

A question which often arises in RTE-B customer classes is "How much memory does my program take?" Or sometimes, "How much memory do I have remaining for application programs?" These questions can be quickly answered by the following procedure.

Refer to the generation record of the RTE-B system. As the system generator runs, it prints out the memory bounds of the modules as they are being loaded (assuming that the "Map Modules" option was chosen) in the sequence LOW MAIN, HIGH MAIN, LOW BASE, and HIGH BASE. For the BASIC Interpreter the LOW BASE memory location is worthy of note; call it BASE. There exist two other locations of importance, SYMTF and FCORE, defined as follows:

SHARING INTERRUPT TABLES WITH A 21MX

Question: How, under DOS IIIB, do I use the WCS 21MX I/O Channels 10 & 11 for other devices, when WCS cards are not needed?

Answer: WCS cards do not generate interrupts; therefore, it is not necessary to make an interrupt table entry for them during a DOS IIIB sysgen. Instead, an interrupt table entry may be made for an alternate device (e.g. magnetic tape) allowing the I/O channels to be shared.

(Continued on page 5)

HOW TO SPOOL IN TCS

The main and segment programs below show the coding that allows TCS to manage spooling concurrently with a user's main application. The main program makes a TCS segment call to the spooling segment. The spooling segment does a TCS I/O without wait call. That call specifies a return address (when the I/O is completed) within the seg-

ment. The instruction following the I/O call is a TCS return to main call. At this point, you are back in the main program, and TCS will handle the scheduling of the spooling segment. That is, control can return to the spooling segment without making a TCS segment call into the segment.

Paul McGillicuddy
HP Cupertino

```

0001      PROGRAM BKICS,3
0002      COMMON ISTK(9,3),IN(6,4),IOUT(30,3),ID(128,4),
0003      -      MEHD(128),LZ,ISPNT,KK,LOC,
0004      -      LBUF(40),I,J,LAST
0005      DIMENSION IPQ(54),ISEG(12),ISEGD(44),IFILE(3),JFILE(3),
0006      -      IDMY(3),IOT(30)
0007      DATA ISEG/2HAT,2HHR,2HS ,2HUS,2HER,2HS ,2HDO,2HNE,2HS ,
0008      -      2HSP,2HOL,2HS /
0009      DATA IFILE/2HBK,2HFI,2HL /,JFILE/2HMH,2HFI,2HL /
0010 C INITIALIZE BUFFERS & TERMINAL COMPLETE COUNTER
0011      CALL TIME (1,1)
0012      CALL BINIT (ISTK,3,9,1)
0013      CALL BINIT (IN,4,6,2)
0014      CALL BINIT (IOUT,3,30,3)
0015      CALL BINIT (ID,4,128,4)
0016      LZ=0
0017 C INITIALIZE TCS
0018      CALL TCS (82,IPQ,5,KK,ISEG,4,ISEGD)
0019 C CLEAR ALL PERIPHERALS AND TERMINALS
0020      DO 20 LC =4,10
0021      CALL TCS (3,LC,LC)
0022      CALL TC SER(1,LC)
0023      20 CONTINUE
0024 C OPEN THE FILES
0025      CALL TCS (84,IFILE,1)
0026      CALL TCS (84,JFILE,2)
0027 C GET THE MESSEGES FROM "MHFIL"
0028      CALL TCS (14,3,MEHD,128,2,1,1)
0029 C OUTPUT MESSEGES TO ALL TERMINALS
0030      ASSIGN 51 TO IRET
0031      CALL TCS (2,20011B,MEHD,30,1,IRET)
0032      CALL TCS (2,20012B,MEHD,30,2,IRET)
0033 C START SPOOLING SEGMENT TO EXECUTE CONCURRENTLY
0034      CALL TCS (8,4)
0035 C SUSPEND AND WAIT FOR COMPLETED TERMINAL
0036      50 CALL TCS (53)
          ;

0001      PROGRAM SPOLS,5
0002      COMMON ISTK(9,3),IN(6,4),IOUT(30,3),ID(128,4),
0003      -      MEHD(128),LZ,ISPNT,KK,LOC,
0004      -      LBUF(40),I,J,LAST
0005      I = 7
0006      J=6
0007 C READ FIRST RECORD FOR END OF FILE
0008      CALL TCS (1,I,LBUF,40,M)
0009      LAST = LBUF(1)
0010 C ESTABLISH RETURN FOR I/O WITHOUT WAIT
0011      ASSIGN 80 TO KRET
0012 C PAUSE WITHOUT WAIT
0013      CALL TCS (1,20077B,1,1,ILU,KRET)
0014 C RETURN TO MAIN
0015      CALL TCS (54)
0016 C START READ/WRITE LOOP
0017      80 CALL TCS (1,I,LBUF,40,M)
0018 C TEST FOR END OF FILE
0019      IF (LBUF(1).EQ. LAST ) GO TO 90
0020      CALL TCS (2,J,LBUF,40,M)
0021      GO TO 80
0022 C LEAVE
0023      90 CALL TCS (53)
0024      99 CALL BKICS
0025      END

```

The spooling segment will execute at priority 0, while the other segments are 1.

This segment reads a card and prints it on the line printer.

A PAUSE without wait is used but is not required, any I/O without wait would do.

SHARING I/O SLOTS (cont.)

Consider the following table:

* EQUIPMENT TABLE ENTRY

17,DVR00	Teleprinter Driver
14,DVR31,D	Disc Driver
10,DVR33,D	WCS Driver
20,DVR02	Punch Driver
16,DVR01	Photoreader Driver
13,DVR12	Line Printer Driver
11,DVR33,D	WCS Driver
10,DVR23,D	Magnetic Tape Driver

/E

* DEVICE REFERENCE TABLE

1 = EQT #?
1
2 = EQT #?
2
3 = EQT #?
2
4 = EQT #?
4
5 = EQT #?
5
6 = EQT #?
6
7 = EQT #?
7
8 = EQT #?
8
9 = EQT #?
3
10 = EQT #?
/E

* INTERRUPT TABLE

11,EQT,8
13,EQT,6
15,EQT,2
16,EQT,5
17,EQT,1
20,EQT,4

/E

The following Equipment Table and Device Reference entries allow reference to the WCS cards (when present in I/O slots 10 & 11) by logical unit numbers 7 and 9:

10,DVR33,D	(EQT = 3)
11,DVR33,D	(EQT = 7)
7 = EQT #?	
7	
9 = EQT #?	
3	

The following Equipment Table and Device Reference entries allow reference to the magnetic tape (when present in I/O slots 10 & 11) by logical unit number 9:

10,DVR23,D	(EQT = 8)
8 = EQT #?	
8	

Additionally, the interrupt table entry:

11,EQT,8

ensures proper handling of magnetic tape interrupts.

Note: Revision 1446 (or later) of DVR23 must be used.

*Jack Howard
HP Cupertino*

RECONFIGURING THE BCS FOR A NEW INTERFACE

Failure to reconfigure BCS after the installation of an interface card results in a common problem — a program appears to become lost in memory without apparent cause. Frequently the interface card is intended to be used with a simple skip-if-flag-is-set driver and the installer incorrectly concludes that reconfiguring BCS is not necessary. Then after writing a program which uses the skip-if-flag-is-set driver, he discovers that the program will not execute properly if a call is made to .IOC., or to the formatter which uses .IOC.

What actually happens is that the program executes satisfactorily as long as the interrupt system is off. However, once a call is made to .IOC., either directly or through a call to the formatter, .IOC. turns the interrupt system on with an STF 0 instruction. The next flag on the interface card causes an interrupt resulting in execution of the contents of the corresponding memory location associated with its select code. Thus the program fails at this point, leaving the programmer with less than the desired result.

This problem can be prevented by configuring BCS appropriately and placing 1067XX (CLC XX where XX is the select code) in select codes containing interface cards not having associated BCS drivers. The CLC will clear control on the interface card preventing the recognition of the interrupt.

*George Taylor
HP Cupertino*

bulletin

CWF SOFTWARE UPDATE INFORMATION

The subscription service is no longer available for the HP software product Course Writer Facility. Customers can receive update packages either by purchasing the system service contract or on a time and materials basis.

*Marilyn Branthwaite
HP Cupertino*

documentation

The following tables list all currently available software manuals, divided into the categories — DOS, SIO, BCS, Timeshare Systems, Languages, and other manuals. Copies of manuals and update packages can be obtained from your local Sales and Service Office. The address and telephone number of the office nearest to you are listed in the back of all reference manuals.

Manuals may also be ordered directly by mail. Simply list the name and part number of the manuals you need on the Corporate Parts Center form supplied at the back of the **Communicator**. If you require an update package (the items marked N/C in the tables) send your request to:

Manual Distribution Center • 11000 Wolfe Road • Cupertino, Ca. 95014

DISC OPERATING SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE
02100-90074	HP 2605A Console Printer Driver	\$ 1.00
02116-91779	Moving-Head Disc Operating System	15.00
02762-90004	HP 2762A Terminal Printer Driver	1.00
02767-90007	DOS/RTE 2767 Line Printer Driver	1.00
02891-90010	HP 2891A Card Reader Driver	1.00
02892-90005	HP 2892A Card Reader Driver	1.00
5951-1366	Cross Reference Table Generator	1.00
5951-1375	Generating DOS-M	1.50
5951-1381	DOS-M/2000C Timeshared BASIC File Handler	1.00
5951-1393	Generating DOS	1.00
5951-1394	2000C File Interface for DOS-M	1.00
12560-90023	DOS and RTE CALCOMP Plotter Driver	1.00
12587-90011	HP 12587B Asynchronous Data Set Interface Driver Reference Manual	5.00
12602-90023	DOS/RTE Mark Sense Drivers Kit 12602B	1.00
12908-90004	HP 12908 Writable Control Store Driver	1.00
12920-90004	HP 12920B Asynchronous Multiplexer Interface Driver Reference Manual	5.00
13024-90010	DOS/DOS-M Magnetic Tape Unit Driver	1.00
24307-90006	DOS-III Reference Manual	20.00
24307-90012	DOS-III Data Communications Drivers	7.50
24307-90018	DOS-III Pocket Guide	3.50
24307-90022	DOS-III Terminal Printer Driver	1.00
24307-90073	DOS-III Standard Drivers	6.00
24376-90001	IMAGE 2000 Data Base Management System Reference Manual	11.00

SOFTWARE INPUT/OUTPUT SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE
02100-90072	HP 2605A Console Printer Driver	\$ 1.00
02116-91760	Teleprinter Driver (LP Compatible) Manual	1.00
02762-90002	HP 2762A Terminal Printer Driver	1.00
02892-90003	HP 2892A Card Reader Driver	1.00
5950-9276	SIO Drum-Disc	1.00
5951-1374	Software Input/Output System Configuration	1.00
5951-1390	Subsystem Operation	1.00
12602-90022	Mark Sense Card Reader Drivers	1.00
12653-90004	HP 2767 Line Printer Driver	1.00
12845-90005	HP 2610A/2614A Line Printer Driver	1.00
12987-90006	HP 2607 Line Printer Driver	5.00
13022-90010	HP 7970 Magnetic Tape Unit Driver	1.00
13029-90010	Magnetic Tape Driver (7-Track)	1.00

BASIC CONTROL SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE
02100-90073	HP 2605A Console Printer Driver	\$ 1.00
02116-9017	Basic Control System Manual	8.50
02762-90003	HP 2762A Terminal Printer Driver	1.00
02892-90004	HP 2892A Card Reader Driver	1.00
5951-1388	Generating HP Basic	1.00
5951-1391	Basic Control System	1.00
12602-90021	Mark Sense Drivers	1.00
12653-90005	HP 2767 Line Printer Driver	1.00
12845-90004	HP 2610A/2614A Line Printer Driver	1.00
12987-90008	HP 2607 Line Printer Driver	5.00
13023-90010	HP 7970 Magnetic Tape Unit Driver	1.00
13026-90010	Magnetic Tape Driver (7-Track without DMA)	1.00
13027-90010	Magnetic Tape Driver (7-Track with DMA)	1.00

TIMESHARED SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE
02000-90048	20856A Timeshared BASIC/2000, Level E, Reference Manual	\$10.00
02000-90049	20856A Timeshared BASIC/2000, Level E, System Operator's Manual	7.00
02000-90073	20854A Timeshared BASIC/2000, Level F, Reference Manual	7.50
02000-90073	Manual Update Package	N/C
02000-90074	20854A Timeshared BASIC/2000, Level F, System Operator's Manual	10.00
02000-90074	Manual Update Package	N/C
5952-4491	20854A Timeshared BASIC/2000, Level F, Pocket Guide	.10

LANGUAGE SPECIALS

PART NUMBER	MANUAL TITLE	PRICE
02116-9014	HP Assembler Manual	\$ 6.50
02116-9015	HP FORTRAN Manual	5.00
02116-9016	Symbolic Editor	4.50
02116-9072	ALGOL Reference Manual	10.00
02116-9077	20392A HP BASIC Reference Manual	15.00
5951-1321	HP FORTRAN IV Reference Manual	6.00
5951-1377	Assembler, FORTRAN, and ALGOL Error Messages	1.00
12907-90010	Implementing the HP 2100 Fast FORTRAN Processor	5.00
24307-90014	Assembler Reference Manual	8.00

ADDITIONAL SPECIALS

PART NUMBER	MANUAL TITLE	PRICE
02000-90055	2000E/2000F IDF Author's Manual	\$ 8.50
02000-90080	HP 2000E to HP 2000F Conversion Guide	1.00
02022-90014	Magnetic Tape Reformatting System Support Utilities	1.50
02100-90129	HP 2100 Microassembler Coding Form	5.00
02100-90140	Decimal String Arithmetic Routines	5.00
02108-90008	Microprogramming 21MX Computers Reference Manual	5.00
02116-91751	Prepare Tape System	2.50
02116-91752	Magnetic Tape System	6.00
02116-91780	2100 Series Relocatable Subroutines	11.00
5951-1352	The Librarian	1.00
5951-1353	Special Purpose Magnetic Tape Loader/CAI English	1.00
5951-1371	HP 2100 Front Panel Procedures	1.00
5951-1376	Basic Binary Loader/Disc Loader, Basic Moving-Head Disc Loader	1.00
5951-1392	Magnetic Tape System	1.00
5951-1397	Commercial Subroutines	15.00
19655-90005	HP 19655B Management/260 Data System Preliminary Reference Manual	10.00
19662-90001	Management/230 Data System Preliminary Reference Manual	5.00
20308-90001	Instructional Management Facility Proctor's Manual	7.00
20308-90003	Instructional Management Facility System Manager's Reference Manual	5.00
20309-90001	Instructional Dialogue Facility Proctor's Manual	10.00
20309-90003	Instructional Dialogue Facility Course Developer's Manual	6.00
20309-90005	Instructional Dialogue Facility Author's Pocket Guide	3.50
20310-90001	HP MATH Teacher's Handbook	5.00
20310-90005	HP MATH Proctor's Manual	5.00
20310-90007	HP MATH Curriculum Guide	20.00
20311-90003	Timeshared Graphics Plotting Package	5.00
20311-90001	Timeshared Graphics for Tektronix Terminals	7.00
20352-90001	Educational Budget and Accounting System — System Overview	10.00
20352-90002	Educational Budget and Accounting System Reference Manual	15.00
20353-90001	Educational Payroll System — System Overview	8.00
24380-90001	HP 2100 Remote Job Entry Processor	3.00
24383-90001	Course Writing Facility	15.00
24384-90001	Student College Information System — System Overview	5.00
24387-90001	Basic Analysis and Mapping Program Manual	18.00
24387-90002	Basic Analysis and Mapping Program Pocket Guide	1.00

training schedule

The schedule for training courses related to HP 2000 and 9600 systems, offered during the months July and August, are presented below. Each issue of the **Communicator** will provide timely information on training to assist you in registering for classes applicable to your system.

MAINTENANCE

COURSE NUMBER	PRODUCT	COURSE LENGTH	DATES	LOCATION
22940A	2100A	10 Days	July 7-18	Cupertino
22940A	2100A	10 Days	July 7-18	Rockville
22940A	2100A	10 Days	Aug. 4-15	Cupertino
22940A	2100A	10 Days	Aug. 11-22	Rockville
22941A	21MX	5 Days	July 14-18	Cupertino
22942A	7900A	5 Days	July 7-11	Cupertino
22942A	7900A	5 Days	Aug. 18-22	Cupertino



SOFTWARE

COURSE NUMBER	PRODUCT	COURSE LENGTH	DATES	LOCATION
22950A	2100 Minicomputer Assembler	5 Days	July 7-11	Rockville
22950A	2100 Minicomputer Assembler	5 Days	July 21-25	Cupertino
22950A	2100 Minicomputer Assembler	5 Days	Aug. 4-8	Rockville
22959A	21MX Assembler	5 Days	July 21-25	Rockville
22959A	21MX Assembler	5 Days	Aug. 18-22	Cupertino
22959A	21MX Assembler	5 Days	Aug. 18-22	Rockville
22960A	21MX Microprogramming	5 Days	Aug. 25-29	Cupertino
22952B	DOS IIIB	5 Days	Aug. 25-29	Rockville
92711B	Real-Time Measurement and Control	10 Days	July 7-18	Rockville
92711B	Real-Time Measurement and Control	10 Days	July 14-25	Cupertino
92711B	Real-Time Measurement and Control	10 Days	Aug. 4-15	Cupertino
92711B	Real-Time Measurement and Control	10 Days	Aug. 4-15	Rockville
92712B	Real-Time BASIC	5 Days	July 7-11	Cupertino
92715B	Batch Spool Monitor	3 Days	July 14-16	Rockville
92715B	Batch Spool Monitor	3 Days	July 21-23	Cupertino
92715B	Batch Spool Monitor	3 Days	Aug. 11-13	Cupertino
92715B	Batch Spool Monitor	3 Days	Aug. 11-13	Rockville
92717A	Distributed Systems	5 Days	July 21-25	Rockville
92717A	Distributed Systems	5 Days	July 28-Aug. 1	Cupertino

HP Training Centers

Training is conducted in the U.S.A. at facilities in Cupertino, California and Rockville, Maryland.

Each Training Center is staffed with professional instructors. Courses are designed such that the student will receive both classroom instruction and practical, hands-on experience. By attending the courses in the recommended sequence for your particular HP system, the student will gain the most beneficial training available to meet the needs of your specific application.

Western Training Center Eastern Training Center

Hewlett-Packard
11000 Wolfe Road
Cupertino, California 95014
(408) 257-7000

Hewlett-Packard
4 Choke Cherry Road
Rockville, Maryland 20850
(301) 948-6370

Accommodations

Students provide their own transportation, meals, and lodging. The Training Registrar will be pleased to assist in securing motel reservations at the time of registration.

Registration

Requests for enrollment in an HP Training Course should be made through your local HP representative. He will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the Training Course, time of class, location and accommodations reserved.

Cancellations

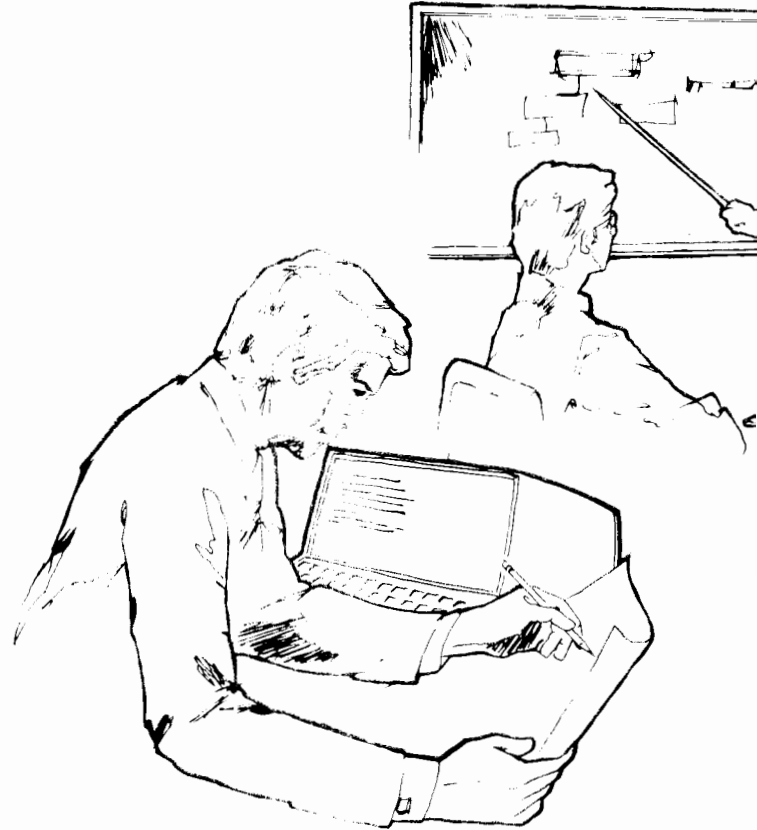
In the event you find you cannot attend a particular class we would appreciate your prompt notification.

featuring — users groups

As an HP Data Systems Customer, you have at your disposal a full array of service organizations and support groups. The **Communicator** is, of course, the newest service offered. It gives our support personnel a direct line to you, making it possible for pertinent information to reach you at regular intervals.

We will be bringing you a closer look at Data Systems service and support organizations from time to time . . . introducing Customer Engineers, System Engineers, Training, and Publications. It is primarily the results of their efforts that we will be publishing each month . . . new products as they are announced, how-to-do-it procedures as they are developed, and system fix-it information devised especially for you the customer.

This month, we've turned our attention to users group organizations as they afford a very special type of service and support. Each users group focuses on a particular area of computer application, such as education, and promotes interaction not only with HP personnel but with other system users. Users groups bring together people with common interest and goals and offer a unique channel for information exchange. We thought you'd like to know more about them.



THE HP EDUCATIONAL USERS GROUP

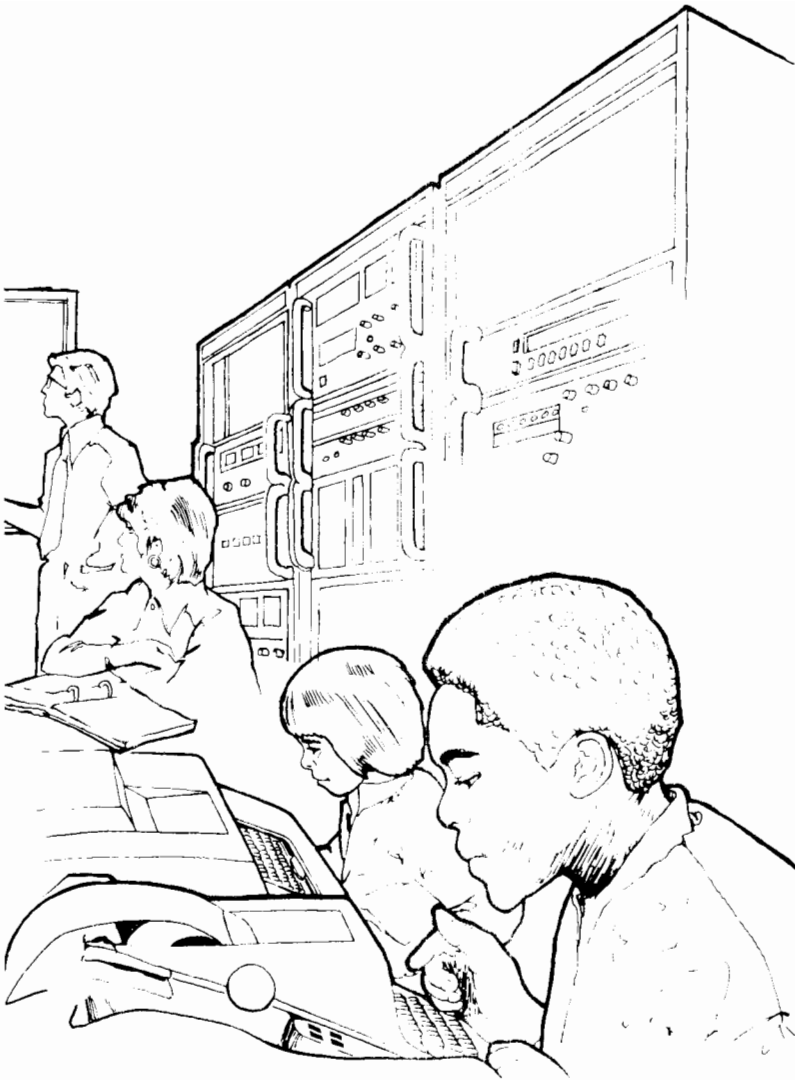
Almost four thousand people are making the computer a viable tool in education. Who are they? The Hewlett-Packard Educational Users Group, a community of people sharing ideas, hopes, and concerns. Using HP computer systems, they are:

- teaching all ages from preschool level to post-graduate
- encouraging the instructional use of computers
- preparing students to understand their role in an increasingly computer-oriented society.

In just four years, this users group has grown from a small group of individuals interested in encouraging the computer's role in education to a worldwide organization with a concentration of members from the U.S., Canada, and Europe. The group is doubling every year as more educators realize the value of using HP computers in their schools.

Why does the Educational Users Group exist?

- to encourage cooperation among the users of BASIC language and HP computer systems in the educational community
- to provide a medium for the exchange of educational software and curriculum materials
- to foster both the instructional and administrative use of computers in education
- to provide information about current research on the use of computers in education
- to help users obtain publicity for their work
- to provide a focal point for solving educational computer users' problems



The continuing focus of the Educational Users Group is the sharing of ideas and experiences. Members feel that they can more effectively promote change, make people aware, and encourage computer use through their users group than as scattered non-communicative individuals.

Since the numerous members of the group are worldwide, most communication is through the Users Group Newsletter. From September to May, users read timely and pertinent information in the Newsletter, the content of which is furnished in large measure by the users themselves.

News is divided into several categories. Reports of specific meetings are found in the section entitled "Users Group News". The column "Group Forum" gives news about individuals and groups of users . . . relating who's doing what where in the field of educational computer application. Other information provided includes news about curriculum development, and new HP hardware and software releases,

and educational resource information. A new feature, the FOCUS issue, which features a specific subject area such as Physics or Social Science has proven to be a wonderful resource for ideas and references.

Hewlett-Packard personnel attend many regional and national conferences of the group and often provide both facilities and refreshments for such meetings. A user coordinator organizes the meetings and arranges for speakers. The Newsletter keeps members informed about such upcoming events.

Users who are located geographically close or who have similar interests are organized into chapters of the main users group. They hold their own meetings, often publish their own correspondence, and have a close association.

If you are not a member, write for further information and a membership form. Faculty or students of institutions that do not own or lease HP systems may receive the Newsletter for a yearly subscription fee of \$6.00. Contact:

HP EDUCATIONAL USERS GROUP
11000 Wolfe Road
Cupertino, Ca. 95014

HP CLEARINGHOUSE

The basic objective of the HP Clearinghouse is to establish and maintain a comprehensive list of educational applications that will run on HP computer systems — both 2000 and 3000 series, as well as selected books and other computer-related documents that would be of potential interest to HP users.

Initially, the Clearinghouse will collect and disseminate information only — the materials themselves would continue to be distributed by the current source agency: the user who developed the package, the HP 2000 or 3000 Contributed Library, or as standard HP supported products. The aim is to have information about items from all such diverse sources available from a single place, that is, the Clearinghouse.

The first catalog will be available around June first. A charge of \$2.00 is being levied to cover printing costs. For information and/or ordering procedures, write us at:

HP CLEARINGHOUSE
Educational Marketing
11000 Wolfe Road
Cupertino, Ca. 95014

THE HP 3000 USERS GROUP

The HP 3000 Users Group, founded in January of 1974, has held three meetings to date — in Palo Alto, Chicago, and Miami. The next meeting is scheduled for October 1975 and will be held in Palo Alto. The meetings are open to all group members and afford an excellent forum for the exchange of new techniques and ideas.

In addition to organizing meetings, the HP 3000 Users Group publishes a periodic newsletter for its membership. The newsletter contains information contributed by group members, often specific problem solving information and original system utilization techniques.

The users group is organized around a Board of Directors composed of a four member Executive Committee and the chairmen of six standing committees: Planning, Technical, Project, Library, Special Interest Groups, and Newsletter Committees.

The Planning Committee is the prime user-HP interface. By means of questionnaires and user reports, the committee coordinates, centralizes, and represents the users and their needs. This committee meets with representatives from Hewlett-Packard every two months to exchange ideas and information. The committee prepares user surveys and correlates responses to the questions for presentation at least once a year to Hewlett-Packard. The Planning Committee also receives and acts upon input from the Technical, Project, and Special Interest Group Committees.

The Technical Committee performs in-depth technical analysis in specific areas of user interest and makes technical recommendations to HP and the users via the Planning Committee. An in-depth investigation of resource allocation, for example, may be undertaken by those customers who have special expertise and experience in resource allocation techniques. Findings may result in recommended techniques for using the system or recommendation as to how the system might be changed.

Project Committee functions include broad-based, non-technical tasks and studies into the development of applications. The results of Project Committee functions are discussed at meetings and passed to HP through the Planning Committee in the form of consensus opinions. These tasks and studies concern the need for human-factors engineering, methods of communication between users, and any other study the Board of Directors deems necessary.

The Special Interest Groups Committee provides a focus for special interests in the applications areas which include manufacturing, education, and business applications across product lines. This committee, composed of subgroups, is an association of those users interested in specific areas of computer application, often where different systems are combined to solve a particular need. This committee affords users maximum interchange on common areas of interest.

The Library Committee establishes contributed software and documentation criteria and also monitors, indexes, and helps coordinate submittals to the HP Contributed Library.

Inquires and submissions may be sent to the Library Committee, as may suggestions for improving the form, and content of contributions. The Library Committee performs a very valuable user function by stimulating and maintaining user contributions.

The Newsletter Committee actively solicits ideas, techniques, and experience for the periodic user communication. This may include reports from the previously mentioned committees as well as information provided by HP personnel.

The current HP 3000 Users Group Board of Directors is listed below. For membership information contact any member of the Board or Alan Mitchell at HP Data Systems Division, 11000 Wolfe Road, Cupertino, Ca. 95014.

Board of Directors

President

Doug Mecham

Hughes Aircraft Company
P.O. Box 3310, Bldg. 601/H219
Fullerton, California 92634
(714) 871-3232, Ext. 3077/3009

Vice-President

Sharon Smith

National Bank of Detroit
P.O. Box 116A
Detroit, Michigan 48232
(313) 225-3699

Secretary

Gerald Schwartz

Hartford Insurance Group
1 Hartford Plaza
Hartford, Connecticut 06115
(203) 547-3669

Treasurer

Tom Harbron

Anderson College
Computing Center
Anderson, Indiana 46011
(317) 644-0951, Ext. 331

Committee Chairmen

Planning

Bill Gates

Longs Drugs
141 North Civic Drive
Walnut Creek, California 94596
(415) 937-1170

Newsletter

Dr. Jim Schwar

Lafayette College
High Street
Easton, Pa. 18042
(215) 253-6821, Ext. 242

Project

Dr. Gary Anderson

McMaster University
Department of Clinical Epid & Biostat
Hamilton, Ontario, Canada L854J9
(416) 525-9140, Ext. 2427

Library

Joe Fleming

P.O. Box 708
Menlo Park, Ca. 94025
(415) 396-2339

Technical

Bruce Weiner

Data Base Management Systems, Inc.
12100 N.E. 16th Avenue
North Miami, Florida 33161
(305) 895-0220, Ext. 136/137

Special Interest Group

Bill Burggrabe

Nooter Corporation
1400 South Third Street
St. Louis, Mo. 63166
(314) 621-6000

software tips

This section of the *Communicator* contains timely how-to-do-it information contributed by HP customers and personnel. The information presented is pertinent to HP 3000 systems only.

You may find just the solution you've been looking for, or merely a better way to accomplish a particular task. If you've found a solution on your own, why not send it to the *Communicator* so that we can share it with everyone.

CLOSING FILES IN FORTRAN

To close a file so that the FORTRAN logical unit number may be used with another file, you should use the procedure UNITCONTROL (described in the Compiler Library manual) with option 8. The ENDFILE statement does not clear the FLUT (FORTRAN logical unit table) entry and so the same file can be referenced at a later time without actually reopening the file. ENDFILE is meant to be used to put an EOF mark on a tape.

*Madeline Lombaerde
HP Cupertino*

CALCULATING THE APPROXIMATE SIZE OF OUTPUT SPOOFLES

An output spoofer contains variable length records; each record has an overhead of 4 words (including carriage control when not embedded in the record). The records are blocked to a maximum of 508 words (4 sectors) per block. Using this information, it is possible to calculate a rough approximation of the disc space required by a particular output spoofer.

Example:

6600 lines, each 132 characters long was written into an output spoofer and required 3772 sectors. This number can be calculated by the following method:

1. # words per record = $\lceil 132/2 \rceil + 4 = 70$ words.
2. # records per block = $\left\lfloor \frac{508 \text{ words/block}}{70 \text{ words/record}} \right\rfloor = 7$ records.
3. # blocks = $\left\lceil \frac{6600 \text{ records}}{7 \text{ records/block}} \right\rceil = 943$ blocks.
4. Disc Space = $943 \text{ blocks} * 4 \text{ sectors/block} = 3772$ sectors.

Where $\lceil \]$ means truncate to nearest whole number less than or equal to the value of the expression and $\lfloor \]$ means round up to nearest whole number greater than or equal to the value of the expression.

Please note that since records normally vary considerably in length, an average record length must be estimated in order to follow these calculations. A certain amount of error will almost always result but at least you'll have a reasonable approximation of the required disc space.

It might also be useful to know the approximate number of sectors required for each page of output. The number of sectors required for one page (60 lines) of records of the specified number of characters are shown below.

CHARACTERS/LINE	SECTORS PER PAGE (60 LINES)
132	36
118	32
72	20
36	12
18	8

Note that the larger the number of pages to be printed per output spoofer the more efficiently the Spooler can utilize disc space. In the calculations given above, 3772 sectors were required to print 110 pages, 60 132-character-lines on each page (6600 lines). If however, we multiplied the number of sectors required for one page, the number would be $110 * 36 = 3960$ sectors (5% error).

*Madeline Lombaerde
HP Cupertino*

CODE SEGMENT ALLOCATION

MPE performance can be improved by allocating the code segments of saved programs semi-permanently into virtual memory. Techniques and hints on allocation are discussed; including how to determine number of CST's left and how many CST entries per subsystem.

Allocation

Whenever a program is :RUN, all of its code segments must be put in virtual memory by entering them into the Code Segment Table (CST) in core. When the program terminates, if no one else is using it, the code segments are removed from the Code Segment Table.

Code segments cannot be used unless they are in the CST. A segment in this state is called "allocated". When it is removed, it is deallocated. The process of allocating and deallocating the code segments consumes a large amount of system resources.

For a program which is going to be run over and over, it would be convenient to permanently allocate its code segments. This is done with the ALLOCATE command. When a program is allocated, it should come into execution faster following a :RUN command.

Allocate/Deallocate Commands

The System Supervisor can allocate programs (and procedures) in virtual memory and deallocate them:

```
ALLOCATE PROGRAM, SPL.PUB.SYS
ALLOCATE, SPL.PUB.SYS
DEALLOCATE, SPL.PUB.SYS
```

Allocation is permanent, *until* next cold load. Then the programs must be allocated again.

Price paid by ALLOCATE is that you use a limited resource — CST entries (MAX=255). When these are all allocated, additional programs cannot be run (or allocated):

```
ALLOCATE PROGRAM, FREE
ERR 228 k
UNABLE TO OBTAIN CST ENTRIES
```

The number of CST entries used by MPE will vary slightly from system to system due to variations in libraries and subsystems. You can obtain the actual number of CST entries used by your system by performing a COOL or COLD start and responding "YES" to the LOAD MAP? request.

Subsystems

The HP 3000 Subsystems (SPL, BASIC, etc.) are program files and can be allocated. This table shows the number of CST entries used by each subsystem (as of 6/15/75):

```
SPL.PUB.SYS      %36(30)
FORTRAN.PUB.SYS %25(21)
BASIC.PUB.SYS   %30(26)
```

(BASIC also pulls in %30 external segments, some of which may not already be allocated)

```
BASICOMP.PUB.SYS %22(18)
```

```
COBOL.PUB.SYS   %34(28)
(COBOL VERSION "A")
(COBOL VERSION "B" %37(31))
```

(Running a COBOL program pulls in the COBOL library, but compiling it does not)

Segmenter (PREP)

```
SEG DVR.PUB.SYS %1(1)
SEG PROC.PUB.SYS %12(10)
```

```
EDITOR.PUB.SYS %17(15)
STAR.PUB.SYS   %21(17)
```

(STAR references %20 external routines.)

```
RJE.PUB.SYS      %5(5)
SYSDUMP.PUB.SYS %5(5)
RPG.PUB.SYS      %31(25)
QUERY.PUB.SYS    %14(12)
```

```
XA2100.PUB.SYS  %5(5)
XL2100.PUB.SYS  %2(2)
FCOPY.PUB.SYS   %5(5)
SORT            %1(1)
```

(SORT VERSION "B")

*Bob Green
M&P Vancouver B.C.*

COBOL CALLABLE INTRINSICS

This article describes a set of interface routines which allows COBOL programs to call MPE intrinsics and routines in the Compiler Library.

HP/3000 COBOL conforms to American National Standard COBOL and is an addition to the 3000 family of languages. COBOL was intended to be a machine independent language. No attempt was made originally to utilize some of the 3000's unique features. But with increased customer sophistication and demand for the ability to access these unique 3000 characteristics, a user oriented technique of calling the system intrinsics has been made available. The method was developed at PROMON ENGENHARIA in Brazil.

COBOL users unfamiliar with the 3000 system intrinsics and the way they work should see the MPE/3000 Operating System Reference Manual (32000-90002), pages 3-6 through 3-10.

The technique developed consists of accessing through standard COBOL CALL statements, a set of interface routines written in SPL to allow the user to access the system intrinsics. The user need have no knowledge of SPL to use these interface routines, but need only follow a set of conventions that follow.

Name

- All procedures in the intrinsic file containing an apostrophe are not callable.
- Other procedures are modified by the interface routine by adding a C at the end as in CHRONOS-CHRONOSC, PRINT-PRINTC, FOPEN-FOPENC.

Parameters

In the intrinsic interface routine each parameter is one of the following structures:

- procedure
- pointer
- label
- simple variable
- array

Procedures, pointers, and labels are handled by the interface routines as one-word quantities. A declaration of:

```
PIC XX
```

will suffice making certain that it is a one word boundary.

Simple variables and arrays can be the following types:

- a. logical
- b. integer
- c. byte
- d. double
- e. real
- f. long

All arrays should be treated the same regardless of type, and sufficient storage to contain results, if any, must be provided. Simple variables must follow a few rules designed to give COBOL full use of the numbers.

1. Integer and logical numbers on the 3000 are one-word quantities but can contain up to five significant digits. In order to handle five digits, you must declare all such parameters as PIC S9(5)-S9(9) COMP. This will use two words of storage with the first one always being zero (or one's for negative numbers), but will allow for numbers in excess of 9,999 to be handled by COBOL. The interface routines make all necessary address corrections.
2. Byte simple variables while rarely used are handled as a character in COBOL.
3. Double and real are two-word quantities on the 3000 but again can go to ten digits which require S9(10) COMP or four words in COBOL the first two of which will always be zero (or one's for negative numbers).

Note: Even if you want to pass a double word of zero it must be declared as S9(10) COMP VALUE 0.

4. Long has no real equivalent in COBOL, but is handled the same way as real or double, requiring four words or eight bytes of storage, the first word of which is always zero.

Summary of parameters:

- a. Procedures, pointers, and labels:

```
PIC XX.
```

- b. Arrays for all types:

```
PIC XX(n) where n varies for each array.
```

- c. Simple variable

```
Byte PIC X  
Integer/logical PIC S9(5) COMP  
Double/real/long PIC S9(10) COMP
```

Condition Code

Since an intrinsic file is used as input for creating the interface routines, it is not possible to determine whether or not a routine modifies the condition code. All routines must be called with a parameter that will return the condition code. This parameter will always be the last parameter passed and should be declared as:

```
PIC S9 COMP
```

The value returned will be 1 for CCG, 0 for CCE, and -1 for CCL.

Typed Procedures

Procedures that are typed (i.e., return a value in the procedure name) must be called with a parameter to contain the answer. This parameter should be declared to meet the same requirements as described for simple variables, and should be positioned in the parameter list just before the condition code parameter.

Variable Procedures

All parameters must be passed to any procedure specifying OPTION VARIABLE

An example of calling the intrinsic READ follows:

```
...  
77 VALUE-DE-PROC PIC S9(5) COMP.  
77 CONDITION-CODE PIC S9 COMP.  
01 PARMS-FOR-READ.  
03 INPUT-BUF PIC X(10).  
03 NUM-CHARS PIC S9(5) COMP VALUE-10  
...  
CALL "READC" USING INPUT-BUF  
NUM-CHARS VALUE-DE-PROC CONDITION-CODE
```

Bit Fields

Since many routines either require bit fields for input (FOP-TIONS and AOPTIONS in FOPEN) or return bit fields on output (FGETINFO, WHO, CHRONOS), a set of bit extract and bit deposit routines have been included in the set of interface routines.

There are routines to handle logical or integer, real or double, and long. All extracts and deposits follow the SPL convention of starting bit, and number of bits and provide wrap-around capability. Bits are numbered beginning on the left with 0 and ending with 15 for integer and logical, 31 for real and double, and 47 for long.

The wrap-around feature means that if the extract or deposit goes beyond the last bit it continues with the most significant bit (0). All routines have the same list of parameters: NUMBER, START, LENGTH, VALUE.

NUMBER
the parameter deposited into or extracted from.

START
the first bit of the deposit or extract.

LENGTH
the number of bits deposited or extracted.

VALUE
the number to be deposited or extracted.

START and LENGTH are both taken modulu 16, 32, or 48 depending on the procedures, and should be declared:

```
START      PIC 99 COMP.
LENGTH     PIC 99 COMP.
```

NUMBER or VALUE should be declared following the rules for simple variables.

Six routines are provided, one for deposit and one for extract for each of the three different length groups.

COBOLEXF	Integer or
COBOLDPF	Logical
COBOLEXFD	Double or
COBOLDPFD	Real
COBOLEXFL	Long
COBOLDPFL	

An example of calling CHRONOS a system intrinsic that returns data and time information follows:

```
77 VALUE-OF-CHRONOS  PIC S9(10)COMP.
77 FIRST-BIT        PIC 99  COMP.
77 NUM-OF-BITS      PIC 99  COMP.
77 JULIAN-DAY       PIC S9(10)COMP.
77 CONDITION-CODE   PIC S9  COMP.

CALL "CHRONOSC" USING VALUE-OF-CHRONOS
                        CONDITION-CODE.
MOVE 7 TO FIRST-BIT
MOVE 9 TO NUM-OF-BITS.
CALL "COBOLEXFL" USING VALUE-OF-CHRONOS
                        FIRST-BIT NUM-OF-BITS JULIAN-DAY.
```

The necessary program and instructions for installation of your system, including a test program are available from the Contributed Library.

*Robert K. Strand
HP Cupertino*

TIPS ON USING THE BASIC/3000 COMPILER

The following are pertinent considerations when compiling BASIC/3000 programs.

MAXDATA Requirements When Using BASICOMP

File and COM buffers are maintained in the DL-DB area and are determined at run time. If MAXDATA is not specified, the initially small DL-DB area won't be large enough and can't be expanded to accommodate the required buffers. One result is that an ASSIGN statement will return error code 5 (NO BUFFER SPACE). Additionally, allowance for DB-Z expansion may be necessary to accommodate growing stacks. The SEGMENTER bases its estimate on the largest stack size estimate for each program. It is therefore recommended that MAXDATA be specified always when COM or files are used and when INVOKEd programs have a lot of non-COM variables.

Deciding on the appropriate MAXDATA size may require some experimentation. The minimum size generally will be 2000 words (decimal). The manual presents guidelines for determining the size: the sum of the record sizes for the maximum number of files open at one time, maximum sum of all COM areas active at one time, 600 words for the DB area, plus allowing for stack expansion required by INVOKE (can use STACK size estimates on compiler listing). Keep in mind that nesting adds to the stack since INVOKE is similar to procedure (SPL) or subroutine (FTN) calls. Try summing the stack sizes plus 700, round upward to the nearest hundred or five hundred. An alternate approach is to request a very large MAXDATA size (20000 or 30000). You may get a message warning "DEFAULT VALUES TAKEN" if the amount you requested exceeds the configuration maximum, but it is only a warning, you won't be using more space than you actually require.

Labeled Entry Points

The following statement presents certain problems for the compiler:

```
CHAIN programname, label.
```

Therefore, when the specified program is compiled (i.e., the one referenced in the CHAIN statement), you must specify all possible labeled entry points; \$COMPILE programname (label₁, label₂, . . . label_n). Determining all labels possibly used as entries can be an arduous task when the label reference is a variable. One solution is to use EDITOR (EDIT/3000) to search for all assignments of that label variable.

String Expressions in CHAIN Statements

When your program contains a statement of the following form:

```
CHAIN string expression
INVOKE string expression
```

that is, the string is non-literal (variable or expression), you can avoid very slow execution times by declaring all possible values for the string expressions (in CHAIN and INVOKE statements) of all subprograms at the time of the non-SUBPROGRAM compile. (BASICOMP generates its own outer block and the user main program is treated like a procedure.) A table of all such declarations is bound to the outer block generated by the compiler. If this isn't done, dynamic LOADPROCs (and possibly UNLOADPROCs) must be done; the time requirements are equivalent to typical CHAIN/INVOKE times in the interpreter.

RLs, SLs, and BASIC

There are no limitations to the use of BASIC statements in RLs (relocatable libraries). SL (segment library) restrictions are:

- a. No COM statements.
- b. An SL procedure cannot CHAIN or INVOKE "backwards" to a BASIC program in the original program file *if* the program name is a literal string. Backward references are allowed only if the program name is a non-literal string expression and declared at compile time.

Program Size

Many production programs used with the Interpreter are large programs and, when compiled, result in a code segment greater than 4K words (typical configuration segment limit). If a major rewrite seems unavoidable, the temptation may be to continue using the interpretive version. Generally, the Interpreter's strengths indicate that it should be used to:

- a. Develop and debug all programs.
- b. Run infrequently used programs.

Compiled programs, on the other hand run faster and put much less load on the system (due to code sharing, smaller data stacks, generally fewer segments). Because of the resulting improved throughput, debugged frequently-executed programs should be compiled.

Therefore, for a large program that already exists, try breaking it into two or three chunks, resolving GOTOs and GOSUBs with CHAINs and INVOKEs. This may not produce as good a result as a complete rewrite, but it will still result in better performance than with the Interpreter.

Programs currently being developed should be written with the compiler in mind: use CHAIN/INVOKE heavily; keep the programs small.

*K. Mintz/M. Lombaerde
HP Cupertino*

bulletins

SCHEDULED FOR OBSOLESCENCE

Two products are in the obsolescence cycle for the HP 3000. MPE-B began the final stages of support March 1, 1975, and all support of this operating system has been terminated June 1, 1975. The MPE-T operating system has yet to announce its formal obsolescence but the paper work is in the mill. Both systems have been replaced by MPE-C which was released late in January of 1975.

*Carolyn Morris
HP Cupertino*



HP 3000 MPE INTERNALS COURSE

The customer, ESL in Sunnyvale California, is involved with various government agencies who as customers demand highly sophisticated applications, some of which are (photographic) image processing and display, and land usage plottage. In some of the applications, non-standard HP 3000 I/O devices are used, hence, special drivers are needed. As ESL is writing its own I/O drivers and is attempting to derive maximum benefit from the HP 3000 system, they saw a need for a greater understanding of the internal activities of MPE.

Customer Engineering Training completed its first HP 3000 MPE Internals Course for Customers during the first week in April. This was a condensed version of the two-week MPE Internals Course that was given to the HP Customer Engineers and Product Specialists during 1974.

The course, as presented, met the requirements and needs of this customer. This course or similarly customized courses are available for customers who also have the need.

All requests for customer training in this area should be directed to:

Tom Lowe
Hewlett-Packard
11000 Wolfe Road
Cupertino, Ca. 95014

software updates

Those programs are now MPE supported. As such, their source is available and two new parts can be ordered from CSC:

Part #32000-1X026 contains the source:

SSAVIOUR.HP32000.SUPPORT
 SSAEDIT.HP32000.SUPPORT
 SRECOVER.HP32000.SUPPORT
 SDPAN.HP32000.SUPPORT
 SLISTLOG.HP32000.SUPPORT
 SLISTEQ.HP32000.SUPPORT

Each issue of the **Communicator** provides you with information pertinent to the status of 3000 software products including the latest software changes and enhancements.

Part #32000-1X027 contains the maintenance files, under HP32000.SUPPORT; namely:

Software updates described in this issue relate to the following products:

MPE 32000C.00.05
 HP 32212A FCOPY/3000
 HP 32104A RPG/3000
 HP 32216A QUERY/3000
 HP 32213 COBOL-A and HP 32213 COBOL-B
 HP 32215 IMAGE/3000

USL Files	Maintenance Files	Job Files	Program Files
USAVIOUR	MSAVIOUR	JSAVIOUR	PSEVIOUR
URECOVER	MRECOVER	JRCOVER	PRECOVER
UDPAN	MDPAN	JDPAN	PDPAN
ULISTLOG	MLISTLOG	JLISTLOG	PLISTLOG
ULISTEQ	MLISTEQ	JLISTEQ	PLISTEQ
USAEDIT	MSAEDIT	JSAEDIT	PSAEDIT

Products described are available through your Customer Engineer, or can be ordered directly via Corporate Parts Center in Mountain View, California.

MPE 32000C.00.05

This article corresponds to the May MIT tap (Date Code 1521) and comprises the official release of MPE 32000C.00.05. The article is organized as follows:

- I Modules Modified for MPE C0.05
- II List of Problems Solved
- III Enhancements to MPE
- IV Release of Supported Utilities
- V Release of Printer-Reader-Punch Subsystem Driver
- VI Documentation Changes
- VII Operational Notes
- VIII Problems

The present release of MPE contains program files of six supported utilities, under PUB.SYS. They are:

SAVIOUR.PUB.SYS
 SAEDIT.PUB.SYS
 RECOVER.PUB.SYS
 DPAN.PUB.SYS
 LISTLOG.PUB.SYS
 LISTEQ.PUB.SYS

I. Module Changes CO.0X

MODULES		1	2	3	4	5	6
INITIAL	0	X		X	X		
SYSDUMP	1	X	X	X			
SEGPROC	2	X	X				
SEG DVR	3						
DISPATCH	4			X			
LOAD	5		X				
MAPP	6					X	
UCOP	7	X					
DEVREC	8						
PROGEN	9	X					
ININ	10					X	
EXIN	11	X	X	X		X	
LOG	12	X					
IOPTRD0	13						
IOPTPN0	14						
IOPLTO0	15						
IOMDISK0	16			X			
IOFDISK0	17			X			
IOTAPE0	18				X		
IOLPRT0	19						
IOCDRD0	20		X				
IOCLTTY0	21						
IOTERM0	22						
IOCDPN0	23						
IOPRPN0	24						N
IOREM0	25						
IOBSC0	26						
IOMDISK1	27	X		N			
PFAIL	30			X	X	X	
FILESYS	50	X	X	X	X	X	
COMM'INT	51	X		X			
STORE/RESTORE	52			X			X

DIRC	53					
ALLOCATE	54		X		X	
DISKSPC	55	X				
MMCORER	56					
MMDISKR	57					
ABORTRAP	58					X
MESSAGE	59					
CROUTINE	60			X	X	
TOUTILITY	61	X		X	X	
TTYINT	62		X	X	X	
PCREATE	63	X				
MORGUE	64			X		
PROCMail	65					
PINT	66					
DATASEG	67	X				
TOPM	68		X			X
CHECKER	69					
UTILITY	70	X	X	X		X
SEGUTIL	71	X		X		
LOADERT	72		X	X		
RINS	73					
JOBTABLE	74	X				
DEBUG	75	X				
NURSERY	76			X		
SYSDPLY	77					
FIRMWARESIM	78	X				
SPOOLING	79			X	X	
SPOOLCOMS	80	X				
MESSAGE CAT				X		

II. List of Problems Solved by MPE C0.05

The intrinsic CTRANSLATE has been modified to incorporate various checking (split stack call, parameters out of bounds, etc.).

The external clock (:SHOWTIME) had a tendency to go fast when the system was loaded. This was corrected.

The **POWER FAIL** message was lost when the console was a GE Terminet. Under this revision, the system sends on "ESC H" sequence to turn the motor on. The message now reads:

```
***POWER FAIL RECOVERY**
```

The "ENQ" is now sent with same parity (odd/even) as the parity made at log on time for 2640 terminal (TERM = 10).

When attempting to purge a file in use, the EOF pointer was reset to 0. It is now left untouched.

A certain number (approximately 400) of line turnarounds on a 202 modem would cause the system to crash. Fixed.

The stack location DB+0 would be zeroed if a line printer file was closed without any data being transferred. Fixed.

Repetitive calls to XCONTRAP to arm and disarm a Control-Y procedure could cause a bounds violation in the program when Control-Y character was entered. Fixed.

III. Enhancements to MPE

Entering Control A on the operator's console has exactly the same effect as pushing on the Console Interrupt button on the panel: the console prompts with an "=" character, awaiting an operator command.

The file system was partially resegmented to include the procedure FREEFSEG BLOCK into segment FILESYS1 in order to reduce intersegment referencing in the frequently used code path of FREAD and FWRITE intrinsics.

Modification of user labels was speeded up by eliminating the need to update the file label. The user label EOF is now maintained internally and updated only when the file is closed.

The system allows a RESTORE operation to continue in the event that a tape read error has occurred on a block other than that containing the file label. When that happens:

1. The disc space allocated for that file is released.
2. An error is recorded for that file.
3. The file is skipped.
4. A TAPE READ ERROR message is output for that particular file, as the cause for not restoring it.

Previously, the same set of events would cause the STORE operation to stop and ignore all remaining files with the "CATASTROPHIC ERROR" message.

IV. MPE Supported Utilities

Along with the MPE C0.05 MIT are released four programs which, in the past, used to belong to the category of "unsupported utilities". Those programs can be found in PUB.SYS on the MIT and are called:

```
SAEDIT.PUB.SYS
LISTLOG.PUB.SYS
DPAN.PUB.SYS
SAVIOUR.PUB.SYS
RECOVER.PUB.SYS
LISTEQ.PUB.SYS
```

They perform the same function of the former "unsupported programs" and it is highly recommended to delete the old versions and keep only the ones hereby released. Those programs are supported by MPE and therefore bugs found in them should be reported to the Customer Engineering organization.

Following is the corresponding documentation for:

```
SAVIOUR/SAEDIT/RECOVER
DPAN
LISTLOG
LISTEQ
```

SAVIOUR/SAEDIT/RECOVER

The set of three programs: SAVIOUR (File Recovery), SAEDIT (Disc Edit) and RECOVER (File Creation) makes possible the recovery of files from an HP 3000 system that has become logically inoperable due to some catastrophic condition such as Cold Load information or System Directory corruption. Such conditions currently reduce the options to bring the system up to only one: RELOAD. It is at this time that the stand-alone cold loadable programs SAVIOUR and SAEDIT can be employed to retrieve files from the disc and copy them to magnetic tape for later recovery.

The File Retrieval program runs stand-alone (independent of MPE) in spite of existing inconsistencies such as invalid volume label or invalid cold load information. Files can be selected for retrieval by the following three options.

1. <file name>.<group name>.<account name>

In this mode, the specified file will be located, if possible, using the System Directory.

2. <logical device number of disc>,
<sector address of file label>

In this mode, the file is located directly by using the <ldev>, <sector address> obtained after a file has STOREd or RESTOREd if the SHOW option had been specified. When a file is located in this way, the <filename group account> from the file label is displayed for verification before the file is allowed to be retrieved.

The locating of files by this method would normally be resorted to because of the invalidity in some way of the System Directory.

3. @.@.@

In this mode, all files described in the System Directory are retrieved using the Directory in locating the files.

A restriction of only retrieving those files accessed since some specified date can be imposed when retrieving files using modes (1) and (3).

The format of each magnetic tape created is that of one or more files, where each file is separated by an EOF mark and the last file by an additional EOF mark. A file consists of 128 word blocks, where the first block is the file label followed by user labels, if any, and data. An individual file on tape will not be allowed to span multiple tape reels. When end of tape is sensed, the tape will be backspaced to the previous EOF mark and the second EOF mark written, a prompt to mount a new output reel will be made and the copying of the file, in its entirety, will continue.

Certain assumptions are made by the File Retrieval program when it is executed.

1. Sector 18 word 14₁₀, the Directory Base Address and word 13₁₀, the Directory Size, must be valid in order to locate the Directory and to perform range checking on addresses referencing the Directory. If a disc error is detected when reading sector 18, then the following dialogue will be entered in an attempt to continue.

CAN'T READ SECTOR 18 OF SYSTEM DISC

THIS CONTAINS DIRECTORY BASE SECTOR AND
LIMIT SECTOR

MAY INPUT VALUES FOR DIRECTORY BASE AND
LIMIT OR HIT CR TO STOP

STARTING SECTOR OF DIRECTORY? <sector address>

NO. OF SECTORS IN DIRECTORY? <Directory size>

2. The Defective Tracks Table, sector 1 of the System Disc, must be valid in order to locate alternate tracks assigned to data residing on discs other than fixed-head discs (tracks are never reassigned on the fixed-head disc).

Failures due to bad tracks when accessing the Directory will be indicated by the following messages as to the reason a file could not be retrieved. The following messages that refer to "ABOVE ERROR" or "ABOVE ERRORS" will be referring to the general disc error message:

DISC { READ }
 { WRITE } ERR ON LDEV#<ldev>
 { SEEK }

STATUS=%<controller status>

ADDR=%<sector> WORD=<word count>

The specific messages are:

SYS/ACT PREFIX CAN'T BE READ
RUN STAND-ALONE DISKEDIT TO INVESTIGATE
ABOVE ERROR DUE TO BAD TRACK — SYS/ACT INDEX

Note: The program cannot continue and will halt because of the above conditions.

ABOVE ERRORS DUE TO BAD TRACK — ACCOUNT ENTRY
ACT/GROUP PREFIX CAN'T BE READ
FOR — <account name>
ABOVE ERROR DUE TO BAD TRACK — ACT/GRP INDEX
ABOVE ERRORS DUE TO BAD TRACK — GROUP ENTRY
GROUP/FILE PREFIX CAN'T BE READ
FOR — <group name>
ABOVE ERROR DUE TO BAD TRACK — GRP/FILE INDEX
ABOVE ERRORS DUE TO BAD TRACK — FILE ENTRY
BAD TRACK — CAN'T READ LABEL

Errors caused by bad addresses in the contents of the Directory or invalid values in sector 18 of the System Disc could be indicated by these messages.

ERROR — TRYING TO READ SECTOR OUTSIDE DIRECTORY

A bad file entry or file label could be indicated by:

```
LABEL COMPARE FAILED
REQUESTED SECTOR NOT ON DISK
LDEV#<ldev> SECTOR #%<sector address>
```

Errors due to invalid definition of the disc configuration can cause the following error messages to be emitted:

```
ERROR — DESCRIPTION FOR DRT 5 WASN'T ENTERED
LDEV#<ldev> NO SUCH TYPE — SUBTYPE
```

Failures in magnetic tape operations will cause the program to go into a hard halt after indicating one of the following detail messages:

```
UNIT WENT OFF LINE
NOT READY INTERRUPT
TRANSFER ERROR
CMD REJECT
TAPE RUN AWAY
TIMING ERROR
TAPE PARITY ERROR
```

followed by one of these operation messages:

```
CAN'T WRITE TO TAPE
CAN'T BACKSPACE
CAN'T WRITE EOF
```

OPERATING PROCEDURES (Stand-Alone File Recovery)

The operator must have a current list of the I/O configuration in order to correctly describe the discs from which files are to be retrieved. The Volume # of a disc is the relative position that a volume definition occupies in the Volume Table. Volume Table entries can be displayed, by their physical ordering, by listing the Volume Table via the INITIAL dialogue.

1. Mount the Cold Load tape containing the two stand-alone programs for File Retrieval and Disc Edit on a tape drive whose DRT is 6 with the unit set to 0.
2. Set the B-register to %003006.
3. Push CPU Reset, I/O Reset, and Cold Load switches on the software panel in that order.
4. Push the RUN switch.
5. Set the B-register to %000001.
(This indicates that the first program on the Cold Load tape is to be loaded. The instructions for creating the Cold Load tape will show that the File Retrieval program is created first.)
6. Push the RUN switch.
7. Set the B-register to %004000.
(This indicates the starting absolute core location that the program is to be loaded.)

8. Push the RUN switch.

(The program is now loaded into core from tape and then begins execution.)

9. The following lines are printed on the console:

```
***** STAND-ALONE FILE RECOVERY (CU.05) *****
ALL NUMERIC INPUT IS DECIMAL — MAY USE OCTAL IF
PRECEDED WITH %
DISC CONFIGURATION SECTION
```

The operator is now prompted with the following dialogue in order to describe the discs on the system. This is repeated until a CR is hit for LDEV#?

```
LDEV#?
DRT#?
UNIT?
TYPE?
SUBTYPE?
```

10. The operator is now prompted as to where to list the names of those files that have been successfully retrieved. Either a line printer or the system console can be chosen.

DRT# of LP? (A CR MEANS USE CONSOLE)

The following message will be output before proceeding to step 11:

```
MOUNT TAPE WITH WRITE RING ON MAG.
TAPE UNIT 0 OF DRT 6.
```

FILE SPECIFICATION SECTION

11. FILE NAME? (OR LDEV#, %SECTOR ADDRESS)
 - a. Enter a particular File Name or,
 - b. Logical Device Number, %Sector Address or where you believe a particular file resides or,
 - c. "@" to indicate all files on the system are to be retrieved.
- If log. device #, %Sector was input, then proceed to step 16 or else proceed as follows:
12. GROUP? } Must use @ in both cases if @ was used
 13. ACCOUNT? } in response to FILE NAME?
 14. DATE? May input a date in form of MM/DD/YY to restrict recovery of files having been accessed from the date entered to the current date.
(CR means ignore the date test.)

15. The File Name retrieved will be printed on the line printer or console along with the logical device number and sector address of the label. Control will return to step 11 if F.G.A. was entered. If @,@,@, was entered, then a listing of all files retrieved will be printed as each file is sorted to tape or the F.G.A. and error message as to why it could not be retrieved will be printed. If @,@,@, was entered, then upon completion, the program prints FINISHED on the console and produces a HALT 0.

16. Since a Ldev#, Sector Address was entered, the following prompt occurs after the label at the specified address has been read:

```
FILE GROUP ACCOUNT — CONTENTS OF LABEL
DO YOU WISH TO RETRIEVE THIS FILE (Y/N)?
```

The File Name, Group, and Account found at the specified address is printed as a check and then you may choose to retrieve the file or not.

17. Control returns to step 11.

ERROR CONDITIONS

Upon serious error conditions, such as:

```
SECTORS OUTSIDE OF DIRECTORY
NO ACCOUNT ENTRIES
BAD TRACK
LABEL COMPARE FAILED
```

you may investigate the Directory and file labels using the STAND-ALONE DISK-EDIT program.

To load this program, perform steps 1-4; in step 5, set the register to %000002 and do steps 6-10.

To get the STAND-ALONE DISK-EDIT instructions, type in HELP or see the instruction description later in this article.

SAEDIT is the stand-alone disc edit and is cold loaded by using the same instructions used to cold load the File Recovery program with the exception of step 5, where instead of setting the B-register to %000001, the B-register is set to %000002.

INSTRUCTIONS FOR STAND-ALONE DISK-EDIT

Note: All input is decimal unless preceded by a '%'. Dump output may be terminated by setting Bit 0 to a '1' in the B-register as the dump is being produced. When the dump is terminated, reset Bit 0 to '0'.

Commands:

<BASE X
where X = the base sector address

<DISC X
where X = the logical device number of the DISC to be selected.

<DUMP X,Y
where X = relative sector starting address
Y = number of sectors to dump

<FORMAT
allows changing the dump format (octal, character, or both) :

<HELP
prints the instructions and commands for using STAND-ALONE DISK-EDIT

<LIST X
where X = DRT of the line printer. (X = 0 means console)

<MODIFY X,Y,Z
where X = relative sector address
Y = starting word within the sector
Z = number of words to modify

(When in the MODIFY mode, two special characters may be input:

'*' — do not change the present value
'/' — terminate the MODIFY operation; that is, no words are modified.)

RESTART
restarts the program to allow re-entering the disc descriptions.

The program RECOVER is a privileged mode program that runs under MPE. This program is used to recreate files back on to the HP 3000 system from the tapes produced by the Stand-Alone File Recovery program. After all the necessary files have been retrieved by the Stand-Alone File Recovery program, the required RELOAD is performed to regenerate the operating system. The account structure must be recreated as well. Next, the RECOVER program should be executed from MANAGER.SYS to restore the files from the recovery tapes. This should be done before restoring all files since the RECOVER program does not delete an existing file and so a file with the same name on the recovery tape will not be restored. Upon completion of the RECOVER program, a complete RESTORE with a KEEP may be performed to return all remaining files to the most up-to-date level.

The building of the Cold Load tapes containing the Stand-Alone File Recovery and Stand-Alone Disk Edit programs is accomplished by the use of the DIAGNOSTIC UTILITY PROGRAM (HP32217A.04.0) named SDUP.PUB.SYS.

An example creation of this Cold Load tape is listed below.

```
=SESSION
:HELLO MANAGER.SYS
SESSION NUMBER = #S3
MON, MAY 5, 1975, 7:30 PM
HP32000C.F0.69

ST/19:30/#S3/LOGON FOR: MANAGER.SYS ON LDEV#3
:RUN SDUP

3000A DIAGNOSTIC UTILITY PROGRAM (HP 32217A.04.0)
ENTER 1 FOR CREATE CPU TAPE
ENTER 2 FOR BUILD STANDALONE TAPE
ENTER / TO TERMINATE

PROGRAM NAME?
SAVIOUR
PROGRAM NAME?
SAEDIT
PROGRAM NAME?
/
?IO/19:31/#S3/32/LDEV# FOR "DT1" ON TAPE (NUM)
=REPLY 32,8
ENTER DRT NUMBER FOR CONSOLE DEVICE
3
ENTER DRT NUMBER FOR LINE PRINTER
0
PROGRAM SAVIOUR ANY CHANGES?
PROGRAM SAEDIT ANY CHANGES?
1 SAVIOUR A
2 SAEDIT A
ENTER 1 FOR CREATE CPU TAPE
ENTER 2 FOR BUILD STANDALONE TAPE
ENTER / TO TERMINATE
/
END OF PROGRAM
:
```



DPAN (CU.05)

The program DPAN produces a simple analysis and a listing of the content of memory as recorded on tape through the core dump procedure. There are cases when the best way to diagnose a software or hardware problem is to take a dump (picture) of the content of core at the time of the crash and to analyze the data so obtained.

The first step of dumping core to a mag tape is performed by a special microcoded function of the system and is initiated by pressing the "SYSTEM DUMP" button (which could as well be labeled "CORE DUMP" in order to avoid confusion with the SYSDUMP command whose purpose is to back the system up). Once the picture of the memory is on the tape, the program DPAN can produce a formatted listing of it which may be analyzed to determine the cause of the problem.

It is highly recommended that those operations be performed each time the system goes down for an unknown reason.

Steps to Produce Core Dump Tape. Once it has been determined that a CORE DUMP should be taken, follow the steps listed below.

- HALT the CPU by pressing RUN/HALT button, if not already in halt state.
- Prepare mag tape drive as follows:
 - Mount and load a "known good" magnetic tape that contains a write ring.
 - Ready drive to ON LINE and select Unit 0.
- Set all toggle switches on both the hardware and software maintenance panels to the NORMAL position (DOWN!!!). These switches are easily identified as those switches that have a label printed both above and below them.
- Set the B Switch Register (on software panel) to 002NNN, where NNN is the device # of the mag tape controller (same as for Cold Load).
- Press I/O RESET button.
- If the machine is not in the normal hardware state (like SYSTEM HALT LIGHT is on), then press CPU RESET button. Otherwise, do not press!!
- Press the SYSTEM DUMP button and watch the magnetic tape for adequate movement of several feet. The CPU should also halt here after the tape has moved several feet.
- Normal Case:* (CPU halted after tape moved several feet. The tape is not rewound.) Press CPU RESET to get machine back to normal state. Rewind the tape manually and run DPAN or send tape with all documentation about system, etc., to the proper person. The system can then be restarted (COLD LOAD, COOLSTART, WARMSTART, or RELOADED, as may be the case.)

Abnormal Case: (CPU did not halt and/or tape did not move an adequate amount.) The dump is essentially lost because the state of the machine has changed.

Some common causes for unsuccessful SYSTEM DUMP execution on first try are:

- Mag tape drive not ON LINE and ready.
- Mag tape Unit 0 not selected.
- Mag tape does not contain a write ring.
- CPU RESET not pressed when SYSTEM HALT light was on.
- FREEZE if INT switch is in ENABLED position.
- Bad magnetic tape itself which causes "Read after Write error".
- IO RESET not pressed after tape was mounted.

Execution of DPAN. The program DPAN can then be run as any ordinary HP 3000 program. Input data to the program is the core dump tape generated above and the output is a formatted printout.

The input file (core dump tape) formal designator is "DPAN-MAST" and defaults to the class "TAPE" and therefore will require operator intervention to select proper drive. The output file (printout) formal designator is "DPANLIST" and defaults to "LP" class if the program is run from a session, to "\$STDLIST" if run from batch job. File equations are necessary to modify those files.

Once in memory and executing, DPAN:

- a. Outputs HP-3000 MEMORY DUMP ANALYZER message on the Job List Device, followed by its own version numbers and date.
- b. Opens all files it will need during execution. At this time, MPE will request operator intervention on DPAN's behalf for opening DPANMAST file. Enter appropriate response on system console.
- c. Reads the SYSTEM DUMP tape and creates a "core-image" file it will use for accessing memory in lieu of the SYSTEM DUMP tape.

Once the SYSTEM DUMP tape has been read and rewound, it may be unloaded and removed from the MT drive.

Note: No switch register is sensed in this version of DPAN.

DPAN Output. Assuming the dump is correct, DPAN will analyze and list the following information:

- a. ID page
- b. Register pages
- c. Low fixed core
- d. System global area.
- e. DRT
- f. CST
- g. DST
- h. PCB
- i. Scheduling queue
- j. Linked memory
- k. Octal main memory dump

Note: The diagnostic of a crashed system through an analysis of the DPAN printout is a delicate operation and should be left to field and factory specialists.

LISTLOG (CU.05)

Purpose.The object of the LISTLOG program is to output, in a readable form, the content of a Log file generated by MPE when the system is running. The output is directed to any device and the type of records listed can be selected.

For more information about the Log file, refer to HP 3000 Manager/Supervisor Manual (Part #32000-90006, Section IX).

Running LISTLOG. Before running the program, two file equations have to be entered to determine the input file (Log file) and the output file (List file). Formal designators for those files are, respectively, "LOGFILE" and "LOGLIST".

The creator of the Log file is "MANAGER.SYS".

The parameter of the RUN command can be used to selectively specify the record types to be output: when PARM is 0, all records will be listed; to suppress the listing of a given record type, say X, set a 1 in bit position (15-x) of the PARM word.

Record Type	X
Logging Error	0
System Up	1
Job Initiation	2
Job Termination	3
Process Termination	4
File Close	5
System Shutdown	6
Spool Input/Output	7

Examples. To list all log records of LOG0034 on line printer:

```
:FILE LOGFILE = LOG0034
:FILE LOGLIST; DEV = LP
:RUN LISTLOG
```

To list Job Initiation and Job Termination records on tape:

```
:FILE LOGFILE = LOG0034
:FILE LOGLIST; DEV = TAPE
:RUN LISTLOG; PARM = %163
```

To list the Log File Error records (if any) to a Terminal (LDEV=21):

```
:FILE LOGFILE = LOG0034
:FILE LOGLIST; DEV = 21
:RUN LISTLOG; PARM = %176
```

Error. The program quits (by call to intrinsic QUIT) with the following conditions:

- QUIT #1 Log file cannot be opened
- QUIT #2 Output file cannot be opened
- QUIT #3 Read failure on Log file
- QUIT #4 Unknown record type (>8)

LISTEQ UTILITY (CU.05)

LISTEQ is used to print file equations and temporary files created during a job/session. The utility does this by accessing the JOB DIRECTORY TABLES (JDT) and listing all current file equations and job/session temporary files.

To use the LISTEQ Utility, issue:

```
:RUN LISTEQ.PUB.SYS
```

Example:

```
:FILE TAPE;DEV=TAPE
:FILE PRINT;DEV=LP
:BUILD INPUT;REC=40,3,F,ASCII;TEMP
:RUN LISTEQ

LISTEQ          (CU.05)

***TEMP FILES
INPUT.PUB.SYS

***FILE EQUATIONS
:FILE TAPE;DEV=TAPE
:FILE PRINT;DEV=LP

END OF PROGRAM
:
```

Note: When no TEMP files or File Equation, such an indication is given as output of the program.

V. Release of Card Reader Punch Driver — (IOPRPN0)

Before using the HP 30119A Card Reader Punch Subsystem, the user should be aware of the following information not presently covered in the operating and reference manual for this device.

When a card reader punch file is opened for simultaneous input and output, Hopper 1 and Stacker 1 are assigned for use by default (unless the programmer specifies otherwise). Thus, if the programmer wishes to read from one hopper and output from the other, he must change the stacker and hopper assignments by interleaving calls to the FCONTROL intrinsic between his read and write requests.

Example:

Statement	Meaning
CRP = FOPEN(, , AOPTIONS, -80, DEV)	Opens reader punch file.
LREAD = FREAD(CRP, TARG, -80);	Reads from Hopper 1, deposits in Stacker 1.
FCONTROL(CRP, %60)	Assigns Stacker 2, Hopper 2.

```
FWRITE(CRP, TARG, -80, 0);           Writes from Hopper 2,
                                     deposits in Stacker 2.
```

```
(Note: In the FOPEN intrinsic, AOPTIONS=%4 and
DEV="CARD".)
```

When a card reader punch has been configured as a job/data accepting device, but the programmer wants to use it to accept data only, he must first ask the console operator to:

1. Issue a =REFUSE ldn command from the console. (The ldn parameter is the logical device number of the card reader punch.)
2. Next, issue an =ABORTIO ldn command from the console.

After these commands have been issued, the programmer is ready to run his/her program.

The card reader punch hardware does not provide for indicating an end-of-file. Thus, when this device is directed to accept data only (as in 2 above), the programmer must indicate the end-of-file through the software. He does this by entering :EOF: on the last record of his/her file, beginning in the first column of that record.

If the programmer wishes to override the default file characteristics of the card reader punch subsystem, particularly when using it with other MPE subsystems, such as FCOPY, he must first issue an appropriate :FILE command.

Example: (Copying from disc to cards)

```
:FILE CRPUNCH;DEV=CARD;ACC=OUT
:RUN FCOPY
FROM=DISCFL;TO=CRPUNCH
```

The card reader punch is a spoolable device. If a user plans to run programs using this device in a spooling environment, he should first read the information about spooling in *MPE/3000 Operating System Reference Manual*.

VI. Documentation Changes

MPE 3000 Operator's Guide (Part #32000-90004)

pg. 4-1

The first sentence should read:

"The console operator initiates communication with MPE/3000 by pressing the CONSOLE INTERRUPT switch on the HP 3000 console panel (. . .), or by pressing simultaneously the control key and the A key of the console."

pg. 9-1

The ****POWER FAIL**** has been changed to:

****POWER FAIL RECOVERY**

(This message was modified to avoid misinterpretation; the same procedure has to be used in order to recover.)

MPE 3000 Manager/Supervisor Manual (Part #32000-90006)

pg. 9-13

The maximum data segment size ever is given in sectors and not in words.

MPE 3000 Reference Manual (Part #32000-90002)

pg. 6-44

The note stating under the FWRITELABEL intrinsic that this procedure does not overwrite old user labels is incorrect.

pg. 6-37

Inversely, it should be stated under FWRITE that this intrinsic cannot access user labels.

pg. 7-9

It should be mentioned under the COPY command (Segmenter) that, once the operation finished, the RBMs in the new segment are found in the reverse order of that in which they were found in the original segment and does not impact on the transfer itself.

VII. Operational Notes

The standard configuration value for the time quantum of the C queue is one second. There are some cases where a smaller value (300 to 500 ms) would get a better response time, without significantly increasing the overhead. Those cases correspond typically to the following conditions:

- An appreciable load of the C queue (seven or more users)
- and
- A mix of highly interactive and computational sessions.

This arrangement would tend to force the interactive response.

There is in the file system a hard coded value of 8K words as an absolute limit for buffer space that any opened file is allowed to have. This (somewhat arbitrary) value was chosen so buffer transactions in and out of core could not jeopardize the throughput of the machine by having to handle segments too big. An attempt to use larger buffers would result in a File System error FCHECK # 57: VIRTUAL MEMORY NOT SUFFICIENT FOR THE FILE SPECIFIED.

The same error (#57) is also used to return the condition that no disc virtual memory was found to allocate buffer space; the latter condition can be overcome by increasing (through SYSDUMP) the virtual memory space.

VIII. Outstanding Problems in MPE

Closing a file with NO REWIND is not implemented.

FSPACE spaces on block rather than records.

Chained SIOs on mag tape do not perform correctly causing transfer larger than 4095 words to fail when record length is undefined.

The character ":" is treated as EOF on \$STDINX.

The commands: LISTACCT, LISTGROUP, and LISTUSER to mag tape can lock the directory indefinitely if the tape is not ready.

Input arguments to BINARY of [\pm] 65336, 65337, 65338, and 65339 do *not* return overflow.

The FOPEN's FORMSG parameter is a byte array: if this array begins on odd byte boundary, the preceding byte is also printed.

Lower case :EOD is not recognized on a data accepting device. Only upper case letters are recognized for that command.

The command :DEALLOCATE of a non-existing program file comes back with ERR 217. It should read ERR 217, 52 where 52 is the corresponding FCHECK number. (The file referenced is not in the system file domain.)

\$STDLIST Allocation Problem. When the "max # of open spoolfiles" (SYSDUMP/INITIAL) is not sufficient to adequately handle all spooling requirements (see below), spooled JOBS to line printers (including :STREAM) may cause "endless" numbers of null list files to be generated in certain situations.

This bug manifests itself as multiple \$STDLIST files for a single JOB, each producing simply a header and trailer with no real JOB output: null files. If the list line printer is spooled, this results in many null spoolfiles (each four sectors); in the case of non-spooled line printers, the headers/trailers are printed immediately — if such non-spooled line printers are offline, the system will crash due to IOQ overflow in this situation. If an open spoolfile is closed during this "resource looping" by some running user, the provoking JOB may finally get launched normally. In this case, the last \$STDLIST file for the JOB will be the true job listing.

This bug can be overcome by increasing the open spooler limit. The recommended value is around 20. A better estimate may be obtained by considering that the *initial* allocation (FOPEN) of a spooled device for *user* access creates a newly opened spooler. On the corresponding FCLOSE, the spooler becomes unopened.

For example:

A SESSION's single access to a spooled line printer results in one open spooler; a spooled JOB to a spooled list device requires at least two — however, each further access to \$STDIN or \$STDLIST within the JOB "reuses" the already allocated ones; each of the JOB's FOPEN's of "LP" though requires another one. Expected concurrency is important because the limit applies to those spoolers open at the same time. An indication that the limit is being pressed is failures on allocations to spooled devices, assuming the spool discs are healthy and have sufficient space.

*Jean-Michel Gabet
HP Cupertino*

HP 32212A FCOPY/3000

This article delineates fixes made in version 00.03 of FCOPY/3000. The following tape may be ordered from CSC in Mountain View:

Part #32212-1X002 Store Format Maintenance

Problems Corrected in FCOPY 00.03

Compare and verify options no longer cause stack overflows.

Compare errors now report the correct byte in error. The first column or byte is 1 and the last is N.

Differing record lengths during a compare now increment the compare error count, report properly, and terminate when the errors equal the number of errors specified in the option (i.e., TO = ---; FROM = ---; COMPARE = 5).

The syntax error messages are now correct.

Problems That Still Exist in FCOPY 00.03

Temp files are closed permanently.

EBCDICIN and OUT conversions still use earlier IBM versions. Later versions will be installed in a subsequent fix.

*M. Roger Jennings
HP Cupertino*

HP 32104A RPG/3000

This article addresses the changes to the HP 32104A RPG/3000 subsystem, and together with MPE-C.00.05 comprises the official release HP 32104A.01.03.

Incorporated in this release level are corrections of the following problems:

- Cross-reference with more than one Image File caused run time error.
- Special File not recognizing end of file.
- Forms alignment not working (no forms alignment message).
- TESTN sometimes aborts if less than three result indicators used.
- Moving constant of 01 or 00 or -01 results in only low order digit being changed.

Note: There was *no* release for fix level .02, therefore, the previous update and fix level was 01.01 and the latest (present) update and fix level is 01.03.

*Wilma Watkins
HP Cupertino*

HP 32213 COBOL-A HP 32213 COBOL-B

This article, along with the release of MPE-C.00.05, comprises the official release of fix level .04 (COBOL-A) and .02 (COBOL-B) for the subsystem HP 32213 COBOL/3000.

Problems Corrected in COBOL-A.01.04.

- 1.0 PICTURE X(0) caused compiler to abort. X(0) is now diagnosed as an error and changed to X(1).
- 1.1 Division yielded incorrect result, e.g., in case of .25/.5, due to a problem in the library.
- 1.2 A FILLER item in the Linkage Section caused a bounds violation or editing problems with currency sign at run time.
- 1.3 The ALPHABETIC test included an erroneous test for a blank in the second character of the field. The library was corrected.
- 1.4 Misspelling of OMITTED (in LABEL RECORD) caused looping in compiler when processing ADD CORRESPONDING statement.

- 1.5 When comparing index-names or index-data items with 0, the result was less than.
- 1.6 The test for data-segment size was corrected. An error message was generated when size exceeded 32K bytes. Also, error #54 was generated instead of error #53.
- 1.7 MOVE of TIME-OF-DAY resulted in BOUNDS VIOLATION at run time.
- 1.8 SEARCH ALL statements with a GO TO statement following the WHEN condition left values on the run time stack, causing a STACK OVERFLOW when executed in a loop.
- 1.9 Multi-level table in DATA DIVISION caused compiler to loop.
- 1.10 Negative multiply with result <10 digits could yield a large positive result due to a problem in the library.
- 1.11 A sequence of commands

```
:COBOL X
:COBOL Y
```

resulted in an automatic \$CONTROL USLINIT at second command. Now, \$OLDPASS will contain the combined USL-file of X and Y (if Y does not contain \$CONTROL USLINIT) after the commands have been executed.

- 1.12 Division of negative value by 1 yielded incorrect positive number due to problem in library.

Problems Corrected in COBOL-B.00.02

- 2.0 PICTURE X(0) caused compiler to abort. X(0) is now diagnosed as an error and changed to X(1).
- 2.1 A FILLER item in the Linkage Section caused a bounds violation or editing problems with currency sign at run time.
- 2.2 Add of a COMP-3 item to a numeric display or COMP item caused error 711 to occur at run time.
- 2.3 Subprogram calls without parameters may cause PREP-error "actual parameters incompatible with formal parameters."
- 2.4 The ALPHABETIC test included an erroneous test for a blank in the second character of the field. The library was corrected.
- 2.5 A (special) forms-message printed an unpredictable additional character before the actual message. This was because the message always started in the second character in a word where MPE requires it to start in the first character of a word.

- 2.6 A move of an alphanumeric table-element to a numeric data-item yielded incorrect result because subscript was ignored.
- 2.7 Misspelling of OMMITTED (in LABEL RECORD) caused compiler to loop when processing ADD CORRESPONDING statement.
- 2.8 When comparing index-names or index-data items with 0, the result was less than.
- 2.9 The test for data-segment size was corrected. An error message was generated when size exceeded 32K bytes.
- 2.10 MOVE of TIME-OF-DAY resulted in BOUNDS VIOLATION at run time.
- 2.11 SEARCH ALL statements with a GO TO statement following the WHEN condition left values on the run time stack, causing a STACK OVERFLOW when executed in a loop.
- 2.12 Incorrect address arithmetic in C'TRAP in the library caused an abort due to invalid intrinsic (PRINT) call (I=65;M=%5(%1)).
- 2.13 Multi-level table in DATA DIVISION caused compiler to loop.
- 2.14 A sequence of commands

```
:COBOL X
:COBOL Y
```

resulted in an automatic \$CONTROL USLINIT at second command. Now, \$OLDPASS will contain the combined USL-file of X and Y (if Y does not contain \$CONTROL USLINIT) after the commands have been executed.

Note: All System Library segments for COBOL-A and COBOL-B are changed in this release.

*Wilma Watkins
HP Cupertino*

HP 32215A IMAGE/3000

This release of IMAGE/3000, HP 32215A.02.00, is corrected and enhanced to reflect the following:

- There are no known bugs outstanding for IMAGE/3000.
- Control-Y procedures have been added to DBUNLOAD and DBLOAD. During execution of either of these programs, striking control-Y (session mode only) will cause the printing of a "progress report", that is, the *approximate* number of entries of the current data set

which have been written to or read from mag tape. The format of the message is:

<CONTROL Y> DATA SET n: X ENTRIES HAVE BEEN PROCESSED

- An additional piece of information may be returned to a calling program's STATUS array by any of the IMAGE procedures except DBOPEN. This is in addition to the information in the fifth through tenth words of STATUS, as described with the release of IMAGE A.01.02. This information is returned whenever these STATUS words are not specifically used for some other purpose. The *sixth* word of the STATUS array now contains:

BITS 7-15: The IMAGE procedure's intrinsic number.

BITS 0-3: The user's current mode of access to the data base, if this can be determined; otherwise, zero.

Summary of Changes to IMAGE/3000 Distribution Files in A.02.00

The *USL*, *SOURCE*, *MAINTENANCE*, and *JOB* files for all IMAGE library procedures have been combined. There is now one of each type. The files are numbered "00". This is a reduction from 32 files to 4.

One new segment has been added to the IMAGE library; it is named DBUS01. It contains procedures used by the IMAGE utilities, *not* by the user.

Install files I01I215A and I02I215A have been removed and replaced by the single install file I00I215A. The INSTRUCT file has been changed to reflect this change.

Items 1 and 3 above mean that there are now a total of 48 files (9 source and 39 maint) for IMAGE instead of the former 77.

Wilma Watkins
HP Cupertino

documentation

The following tables list all currently available HP 3000 software manuals. Copies of manuals and update packages can be obtained from your local Sales and Service Office. The address and telephone number of the office nearest to you are listed in the back of all reference manuals.

Manuals may also be ordered directly by mail. Simply list the name and part number of the manuals you need on the

Corporate Parts Center form supplied at the back of the **Communicator**. If you require an update package (the items marked N/C in the tables) send your request to:

Manual Distribution Center
11000 Wolfe Road
Cupertino, Ca. 95014

MPE/3000 MANUALS

PART NUMBER	MANUAL TITLE	PRICE
03000-90096	Multiprogramming Executive General Information Manual	\$ 4.00
32000-90002	32000C MPE/3000 Reference Manual	19.50
32000-90004	32000C MPE/3000 Console Operator's Guide	7.00
32000-90006	32000C MPE/3000 System Manager/System Supervisor Manual	13.00

LANGUAGE MANUALS

PART NUMBER	MANUAL TITLE	PRICE
03000-90002	SPL/3000 Reference Manual	\$ 6.00
03000-90003	SPL/3000 Textbook	7.50
03000-90003	SPL/3000 Textbook Update Package #1	N/C
03000-90007	FORTRAN/3000 Reference Manual	8.50
03000-90008	BASIC/3000 Interpreter Reference Manual	10.00
03000-90014	COBOL/3000 (Version A) Reference Manual	13.00
03000-90025	BASIC for Beginners	4.50
03000-90047	Cross Assembler for 2100 Computers Reference and Application Manual	17.00
03000-90050	HP 32101A BASIC/3000 Interpreter Pocket Guide	2.50
32103-90001	BASIC/3000 Compiler Reference Manual	3.50
32104-90001	RPG/3000 Compiler Reference and Application Manual	20.00
32104-90003	RPG Listing Analyzer	.50
32213-90001	COBOL/3000 (Version B) Reference Manual	20.00

ADDITIONAL MANUALS

PART NUMBER	MANUAL TITLE	PRICE
03000-90009	HP 3000 Compiler Library Reference Manual	\$ 8.50
03000-90010	HP 3000 Scientific Library Reference Manual	5.00
03000-90011	STAR/3000 (Statistical Analysis Routines) Reference Manual	4.00
03000-90012	EDIT/3000 Reference Manual	4.00
03000-90015	HP 3000 Symbol Trace Reference Manual	4.00
03000-90053	SORT/3000 Reference Manual	4.50
03000-90064	FCOPY/3000 Reference Manual	5.00
03000-90107	HP 3000 Cross Loader for HP 2100 Computers	11.00
30130-90001	2780/3780 Emulator Subsystem Reference and Application Manual	10.00
30300-90002	HP 30300A Programmable Controller Reference and Application Manual	12.00
32215-90001	IMAGE/3000 Reference Manual	7.00
32216-90001	QUERY/3000 Reference Manual	7.00
32900-90001	Student Information System Reference Manual	18.00
32900-90002	Student Information System—System Overview	7.00
36995-90013	IBM 1130/1800 to HP 3000 FORTRAN Conversion Guide	6.00

training schedule

The schedule for software training courses related to the HP 3000 offered during the months July and August is provided below. Each issue of the **Communicator** will provide timely information on training to assist you in registering for classes applicable to the operation of your system.

SOFTWARE COURSES

COURSE NUMBER	PRODUCT	COURSE LENGTH	DATES	LOCATION
22956A	3000 IMAGE	5 Days	July 21-25	Cupertino
22956A	3000 IMAGE	5 Days	Aug. 25-29	Cupertino
22962A	3000 Commercial/Business User	5 Days	July 7-11	Cupertino
22963A	3000 Scientific/Engineering User	5 Days	July 21-25	Rockville
22963A	3000 Scientific/Engineering User	5 Days	Aug. 4-8	Cupertino
22964A	3000 System Management	3 Days	July 14-16	Cupertino
22964A	3000 System Management	3 Days	July 28-30	Rockville
22964A	3000 System Management	3 Days	Aug. 11-13	Cupertino

HP Training Centers

Training is conducted in the U.S.A. at facilities in Cupertino, California and Rockville, Maryland.

Each Training Center is staffed with professional instructors. Courses are designed such that the student will receive both classroom instruction and practical, hands-on experience. By attending the courses in the recommended sequence for your particular HP system, the student will gain the most beneficial training available to meet the needs of your specific application.

Registration

Requests for enrollment in an HP Training course should be made through your local HP representative. He will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the Training Course, time of class, location and accommodations reserved.

Accommodations

Students provide their own transportation, meals, and lodging. The Training Registrar will be pleased to assist in securing motel reservations at the time of registration.

Cancellations

In the event you find you cannot attend a particular class we would appreciate your prompt notification.

Western Training Center

Hewlett-Packard
11000 Wolfe Road
Cupertino, California 95014
(408) 257-7000

Eastern Training Center

Hewlett-Packard
4 Choke Cherry Road
Rockville, Maryland 20850
(301) 948-6370

Although every effort is made to insure the accuracy of the data presented in the **Communicator**, Hewlett-Packard cannot assume liability for the information contained herein.

Prices quoted apply only in U.S.A. If outside the U.S., contact your local sales and service office for prices in your country.

direct mail order form

To expedite your order for software, manuals, or other materials described in this publication, use this form to order directly from the Corporate Parts Center in Mountain View, California.

1. Enter your name, address, customer reference number, and tax exemption information.
2. List the item or items you want by part number and description.
3. Compute the amount due and enclose a check or money order payable to Hewlett-Packard.

If you need assistance in placing your order, contact your local HP Sales Office.

Note: Update packages and manual supplements cannot be ordered from the Corporate Parts Center. Direct your request to:

Manual Distribution Center
Hewlett-Packard
11000 Wolfe Road
Cupertino, Ca. 95014



CORPORATE PARTS CENTER Direct Mail Parts and Supplies Order Form

SHIP TO:

NAME _____ CUSTOMER REFERENCE # _____
 COMPANY _____ TAXABLE*? _____
 STREET _____ CITY _____ STATE _____ ZIP CODE _____

Item No.	Check Digit	Part No.	Qty.	Description	List Price Each	Extended Total

Special Instructions					Sub-total	
Tax is verified by computer according to your ZIP CODE. If no sales tax is added, your state exemption number must be provided: # _____ If not, your order may have to be returned.					Your State & Local Sales Taxes	
Check or Money Order, made payable to Hewlett-Packard Company, must accompany order.					Handling Charge 1 50	
When completed, please mail this form with payment to:					TOTAL	

HEWLETT-PACKARD COMPANY
 Mail Order Department Phone: (415) 968-9200
 P.O. Drawer #20
 Mountain View, CA 94043

Most orders are shipped within 24 hours of receipt. Shipments to California, Oregon and Washington will be made via UPS. Others shipments will be sent Air Parcel Post, with the exception that shipments over 25 pounds will be made via truck. No Direct Mail Order can be shipped outside the U.S.

Computer Systems **COMMUNICATOR**
 Published by: HP Data Systems Division
 Systems Engineering Department
 11000 Wolfe Road
 Cupertino, Ca. 95014

JUNE 15 1975