HEWLETT hp PACKARD

computer systems

COMMUNICATOR

3000

SERIES I 1701 SERIES I 1709 SERIES II 1701 SERIES II 1709

Issue No. 13

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

TABLE OF CONTENTS

EDITOR'S NOTE

Notice anything different about the cover? Things have changed slightly - the COMMUNICATOR has been divided into three separate publications for the HP 1000/2000/3000 - and for a good reason . . . . . YOU!

Now, each issue of the COMMUNICATOR 3000 will deal specifically with HP 3000 related products. The main objective of the publication remains the same however: TO HELP YOU GET THE MOST FROM YOUR HEWLETT-PACKARD COMPUTER SYSTEMS.

Two 1701 M.I.T.'s, one for each Series, are in this issue, since these were not previously documented. Our delivery policy is to provide you with M.I.T. documentation prior to receiving updates from your C.E., as is the case with the 1709 M.I.T.'s.

Notice anything different about the text? To back up this delivery policy, a number of steps have been taken to speed up publication and assure on-time information. One visible step is using the EDITOR/3000 subsystem and a line printer for text preparation.

In this issue of the COMMUNICATOR 3000, you will find new product announcements, software updates, current documentation, and miscellaneous software information.

The COMMUNICATOR 3000 is a publication designed to meet the informational needs of HP Computer Systems users. If you have new ideas, or can suggest changes in the content or format of the COMMUNICATOR 3000, please contact the editor.

Address your correspondence to:

EDITOR
COMPUTER SYSTEMS - COMMUNICATOR 3000
HP General Systems Division
5303 Stevens Creek Boulevard
Santa Clara, CA 95050

INTRODUCING THE HP 3000 SERIES I

by Dave Sanders
HP General Systems

In April 1977, Hewlett-Packard announced an additional entry level model HP 3000, the HP 3000 Series I. This system contains an HP 3000 Series I CPU, 128K-bytes of core memory, HP's new 50 megabyte 7920A disc drive, a 1600 BPI tape, a 2640B CRT system console, and includes the MPE-C operating system. The Series I system is fully upgradeable to Series II systems with semi-conductor memory. Upgraded systems are fully expandable to 512K-bytes, allowing the Series I customer ultimate access to the full range of Series II performance and capabilities.

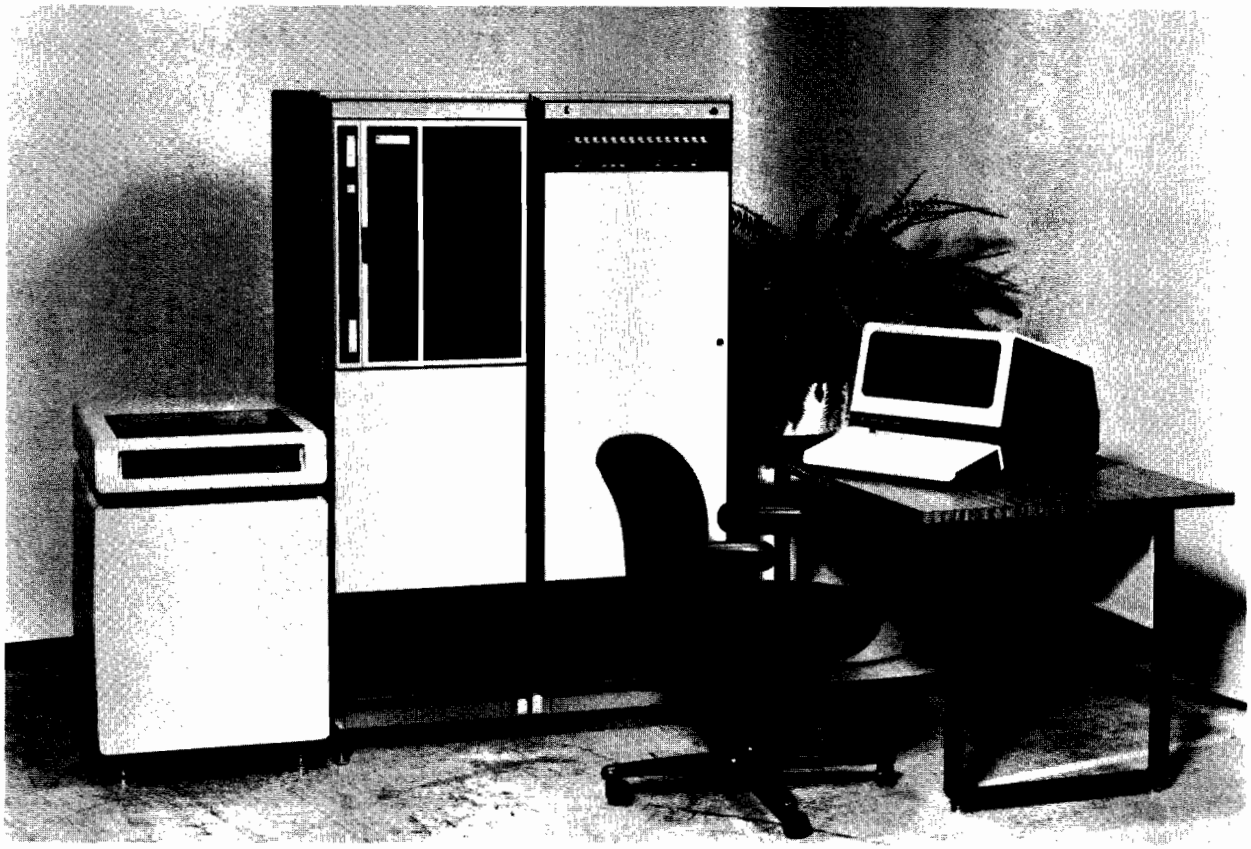
The HP 3000 Series I is the result of an integrated engineering and manufacturing project to utilize previously employed mainframe components returned to HP when HP 3000CX and pre-CX systems are upgraded to Series II Systems. These components were modified to include a new printed circuit CPU backplane and card cages, combined with new peripherals and peripheral controllers, and integrated to form the Series I system. By HP calculation, 85% of the Series I system is of new manufacture. The Series I system includes the standard Hewlett-Packard 90-day warranty.

Customers who own HP 3000CX or pre-CX HP 3000 systems can benefit from this introduction as well. The new 7920A disc drive, which is much faster than its predecessor, the 2888A made by ISS, is available to owners of pre-Series II HP 3000's. It uses the same controller as the 7905, and may be intermixed with 7905's on the same controller. The controller supports up to 8 drives in any combination of 7920's and 7905's. Customers who already have 7905's on pre-Series II machines can therefore add 50 megabytes of disc storage very easily. If your system does not have a 7905/7920 controller, it may be added to your configuration. However, only one other disc controller can exist in a pre-Series II system concurrently with the 7905/7920 controller.

The printed circuit CPU backplane with new card cage will be installed in pre-Series II systems under the terms of the HP service contract if the system develops CPU problems that are isolated to the CPU backplane area. Replacing the existing backplane with the newer unit will not provide any benefit unless the system is experiencing backplane problems.

At the same time, Hewlett-Packard is also announcing a new supported software product, INDEX/3000, which provides an indexed sequential file management capability for the HP 3000 Series I. This product makes it possible for the programmer to create and maintain disc files whose records can be accessed by the value of a single key. INDEX/3000 provides the user with the functional capabilities found in R'ISAM, an unsupported software capability used by some installations.

INDEX/3000 is callable from RPG, COBOL, FORTRAN, and SPL. The user interface is identical to R'ISAM when called from RPG; when called from COBOL, FORTRAN, or SPL, the interface is similar (but not identical) to KSAM/3000. KSAM/3000 is only available on HP 3000 Series II systems since it is integrated internally into MPE-II, the Series II operating system. Utilities exist in KSAM/3000 to convert all INDEX/3000 files and R'ISAM files to KSAM/3000 files if the installation upgrades to a Series II and acquires KSAM/3000. INDEX/3000 will be supported on the Series II for those customers who choose not be acquire KSAM/3000. All users of HP 3000CX or pre-CX systems can add INDEX/3000 to their systems if they desire.



The new HP 3000 Series I combines field-proven mainframe circuitry with new components and peripherals into a computer system equivalent in power to the Series II Model 6 -- yet at an entry level price of $75,000. Shown above is the minimum configuration Series I, including central processor, 16-port asynchronous terminal controller, 7920 disc drive and 7970 magnetic tape drive subsystems, and 2640B system console.

SERIES I 1701 SOFTWARE UPDATE

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES I
****

CONTENTS OF M.I.T. DATE CODE 1701

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE	SUPPORT ACCOUNT FILE CONTAINING CHANGES**
*MPE	32000C	00.14	1701	N00N000C.HP32000.SUPPORT
SPL	32100A	06.02	1621	
BASIC	32101B	00.05	1646	
*FORTRAN	32102B	00.06	1701	N00N102B.HP32102.SUPPORT
BASIC COMPILER	32103B	00.02	1646	
*RPG	32104A	03.02	1701	N00N104A.HP32104.SUPPORT
BUILDINT	32150A	03.01	1623	
*EDITOR	32201A	06.00	1701	N00N201A.HP32201.SUPPORT
STAR	32204A	01.00	1603	
SCIENTIFIC LIBRARY	32205A	00.02	1430	
SDM	32210A	05.00	1508	
*COMPILER LIBRARY	32211C	04.05	1701	N00N211C.HP32211.SUPPORT
*FCOPY	32212A	01.03	1701	N00N212A.HP32212.SUPPORT
*COBOL	32213B	02.04	1701	N00N213B.HP32213.SUPPORT
*COBOLC	32213C	01.02	1701	N00N213C.HP32213.SUPPORT
*SORT/MERGE	32214B	01.03	1701	N00N214B.HP32214.SUPPORT
*IMAGE	32215A	04.02	1701	N00N215A.HP32215.SUPPORT
*QUERY	32216A	03.03	1701	N00N216A.HP32216.SUPPORT
TRACE	32222A	03.00	1610	
*XA2100	32223A	01.01	1701	N00N223A.HP32223.SUPPORT
XL2100	32226A	02.00	1636	
CALCOMP PLOTTER	30126A	00.01	1640	
2780/3780 EMULATOR	30130B	02.00	1646	
PROG CTRLR/BCS	30100A/ 30361A	00.01	1512	
*PROG CTRLR/RTE-C	30301A/ 30361A-1	00.02	1701	N00N301A.HP30301.SUPPORT

* Updated/changed in this M.I.T.

** Changed files not documented in the following pages appear in Series II 1701 documentation.

MPE 32000C.00.14

DATE CODE 1701, N00N000C.HP32000.SUPPORT

A. MODULES MODIFIED

MODULE		CHANGE HISTORY													
NAME	NO	C.00.14													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
INITIAL	0	X		X	X			X	X	X	X	X	X	X	
SYSDUMP	1	X	X	X			X		X	X			X		X
SEGPROC	2	X	X				X			X	X				
SEGDVP	3												X		
DISPATCH	4			X			X				X	X	X		
LOAD	5		X										X		
MAPP	6					X									
UCOP	7	X													
DEVREC	8														
PROGEN	9	X							X	X	X		X		
ININ	10					X		X			X	X		X	
EXIN	11	X	X	X		X	X	X		X		X			X
LOG	12	X									X				
IOPTRD0	13														
IOPTPN0	14						X		X						
IOPLOT0	15														
IOMDISK0	16			X				X	X	X					
IOFDISK0	17			X				X	X						
IOTAPE0	18				X				X						
IOLPRT0	19												X	X	
IOCDRD0	20		X				X								
IOCLTTY0	21														
IOTERM0	22											X			
IOCDPN0	23														
IOPRPN0	24					*	X					X	X		
IOREM0	25														
IOBSC0	26														
IOMDISK1	27	X		*				X	X	X	X				
PFAIL	30			X	X	X									
FILESYS	50	X	X	X	X	X	X	X	X	X	X	X	X	X	X
COMM'INT	51	X		X			X					X	X		X
STORE/RESTORE	52			X	X			X	X	X			X		
DIRC	53									X				X	
ALLOCATE	54		X		X				X			X	X		
DISKSPC	55	X													
MMCORER	56						X		X			X			
MMDISKP	57											X			X
ABORTRAP	58						X	X		X		X		X	
MESSAGE	59							X		X	X	X			
CROUTINE	60			X	X						X	X			

IOUTILITY	61	X		X	X			X	X	X				X
TTYINT	62		X	X	X			X		X				
PCREATE	63	X									X			
MORGUE	64			X						X	X			
PROCMAIL	65													
PINT	66						X		X	X		X		X
DATASEG	67	X								X				
IOPM	68		X			X						X	X	
CHECKER	69													
UTILITY	70	X	X	X		X					X			X
SEGUTIL	71	X		X				X		X			X	
LOADER1	72		X	X					X		X			
RINS	73						X						X	
JOBTABLE	74	X												
DEBUG	75	X												
NURSERY	76			X										
SYSDPLY	77						X							X
FIRMWARESIM	78	X								X	X	X		
SPOOLING	79			X	X			X	X	X	X	X		X
SPOOLCOMS	80	X					X		X		X		X	
MESSAGE CAT	--			X			X	X	X	X	X		X	X

*New

B. SUPPORTED UTILITIES MODIFIED

UTILITY	CHANGE HISTORY									
	5	6	7	8	9	10	11	12	13	14
DISKEDIT		*								
DPAN		*						X		
FREE			*							
LISTDIR			*		X					
LISTEQ		*						X		
LISTLOG		*								
PATCH			*							

RECOVER	*	X			
SAEDIT	*		X		
SAVIOUR	*		X		
SLPATCH		*			

*NEW

C. KNOWN PROBLEMS

1. Closing a tape file with NO REWIND is not implemented.

2. FSPACE spaces tape files by blocks rather than by records.

3. Chained SIOs on magnetic tape do not perform correctly, causing transfer of blocks larger than 4096 words to fail if the record format is variable or undefined.

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

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

6. If the FORMSG parameter of FOPEN begins on an odd byte boundary, the preceding byte is also printed.

7. Lower case :eod is not recognized as an end-of-file on data accepting devices.

8. Issuing a :DEALLOCATE command for a non existent program file returns an ERR 217. The error should be ERR 217,52. The 52 is the file system error number returned from FCHECK.

9. DEBUG break points cannot be set in dynamically loaded procedures except by specifying the physical CST numbers.

10. When DPAN finds that the PCB table has been filled, it prints the erroneous messages "INVALID FIRST UNASSIGNED ENTRY" and "INVALID BACKWARD SUBQUEUE POINTER" even though there is no error in the PCB.

11. When the maximum number of open spoofles is not sufficient to handle all spooling requirements, spooled JOBs may cause endless numbers of null list files to be generated. This bug manifests itself as multiple $STDLIST files for a single JOB, each producing only a header and trailer. If the line printer is spooled, this results in many null spoofles, each using four sectors of

disc space. If the line printer is not spooled, these null spoofles will begin printing immediately unless the printer is not ready. In this case, the system will crash due to an IOQ overflow. If an open spoofle is closed during this resource allocation loop, the job may be launched normally. In this case, the last spoofle for $STDLIST will be the true job listing.

This bug can be overcome by increasing the maximum number of open spoofles. The recommended value is 20, but a more exact figure can be found by examining the usage of your system. Each initial allocation (FOPEN) of a spooled device uses one open spoofle. When the file is closed (FCLOSE), the spoofle becomes unopened.

For example:

A SESSION's single access to a spooled line printer requires one opened spoofle; a spooled JOB requires at least two, one for $STDIN and one for STDLIST. Each additional access to a file of device class LP requires an additional open spoofle.

- One indication that the limit is being reached is allocation failures for spooled devices.

12. The line printers 2613A, 2617A, and 2618A may intermittently report a unit failure condition to the I/O driver that will abort the print operation. This condition has been observed when the unit is brought ONLINE after being placed OFFLINE while printing.

COMPILER LIBRARY HP32211C.04.05

DATE CODE 1701, N00N211C.HP32211.SUPPORT

CORRECTIVE SOFTWARE CHANGES

1. When using the TAB operator (Tnnn) in a FORTRAN FORMAT statement and tabbing backwards then forward, but not to the end of the line, all data following the very last item entered was lost.

2. This version of the COMPILER LIBRARY only supports three word long floating point operations.

COBOL/3000 HP32213B.02.04

DATE CODE 1701 N00N213B.HP32213.SUPPORT

A. ENHANCEMENTS

1. Main program segments no longer have to begin at PB+0 in their respective segments.

2. File specified in an FD entry can be locked/unlocked with the use of a new parameter on the $CONTROL command and/or a new parameter in the SELECT clause:

   a. $CONTROL LOCKING will allow all files to be locked/unlocked. The command may appear anywhere in the source program.

   b. A ,L after the forms message parameter in the SELECT clause will allow that particular file to be locked/unlocked.

   If a $CONTROL LOCKING command is present, it will override any SELECT clauses without the locking parameter.

The following procedures have been added to the COBOL Library in version 4.05:

3. COBOLLOCK. This procedure has three parameters and may be called directly from a COBOL program to lock a file. The second and third parameters are both PIC S9(4) USAGE COMP. Parameter 1 is the file-name from the FD entry of the file to be locked. Parameter 2 corresponds to the lockcond parameter of the FLOCK intrinsic and Parameter 3 is used to return the condition code from FLOCK. It will be a -1 (CCL), 0 (CCE), or +1 (CCG).

4. COBOLUNLOCK. This procedure has two parameters and may be called directly from a COBOL program to unlock a file. The first parameter is the file-name from the FD entry of the file to be unlocked. The second parameter is a PIC S9(4) USAGE COMP item in which the condition code from FUNLOCK will be returned. The same condition code values as for COBOLLOCK apply.

B. CORRECTIVE SOFTWARE CHANGES

1. Level 77 entries with USAGE clauses were being erroneously detected as errors if the last group item in the File Section had a different USAGE clause.

2. If an Undefined Procedure Name error exceeded the limit of allowable errors, the procedure names would not be listed with the error message.

3. The ASCII/Binary parameter of the SELECT clause did not work properly.

4. The construct IF condition THEN NEXT SENTENCE without an ELSE clause caused an Error 211--Undefined or Multiply Defined Internal Label.

5. An 01 Level REDEFINES clause in the Linkage Section did not always work properly.

6. Moving a table element to an alphanumeric edited item did not work.

7. A divide in a COMPUTE statement in which the divisor was an expression of more than one COMPUTATIONAL item did not work properly.

8. A SORT statement nested within a conditional clause (such as an AT END clause) caused an Error 211--Undefined or Multiply Defined Internal Label.

9. Several problems with decimal point alignment in REMAINDERS were fixed.

10. Non-dynamic subprograms could not be put into an RL.

11. Moving a non-dynamic subprogram from the last segment in a USL file into another segment produced an Invalid Program File.

12. Compiling a subprogram with secondary entry points with names similar to the subprogram name into an existing USL file would not always inactivate previous secondary entry points.

13. A program with more than 1000 VALUE clauses would not PREP.

14. A PICTURE clause containing floating minus (or plus) signs to the right of the decimal point truncated the last digit.

15. A BLOCK CONTAINS n CHARACTERS clause without a RECORD CONTAINS clause did not compute the proper blocking factor.

16. Decimal point alignment was not performed on a move of a numeric literal item to a numeric item.

17. Moving a value to TALLY in a subprogram did not work.

18. A non-standard level number was not always handled properly. Example: an 01 level followed by an 05 followed by an 02.

C. KNOWN PROBLEMS

A READ statement in which the file-name is omitted causes the compiler to abort.

PROGRAMMABLE CONTROLLER HP 30301A.00.02

DATE CODE 1701, N00N301A.HP30301.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

1. DNSOUSER routine in SL.PUB.SYS was modified to transmit only the significant words in an absolute record, rather than the actual number of words in the record. This was done by using the length in the left 8 bits of the first word and adding three. NOTE: The XL2100 HP32226A.02.00 DATECODE 1636 or subsequent versions must be used.

2. XRTGEN in SL.PUB.SYS was changed, the interrupt table was not being handled correctly when a Power Fail routine was included as any entry but the last one.

   Type 7 utility subroutines were not being appended to each program which called them. They were being loaded only once. This was corrected with the addition of two new commands: LOCKSYS and CLEARPROG. When a type 7 routine is referenced by a program it is appended to the program. CLEARPROG then clears out the program and entry points. A subsequent program which references the same routine results in the appending of XL2100A.02.00 or later. Earlier versions will abort with an invalid command message.

   A file name with an entry point list was specified as a SAVE file in the format: FILE(E1,E2,...,EN) with a blank inserted into the character position preceeding the first comma. This caused an erronous 'NOT LST ENTRY' to be printed by XL2100.

   The INSTRUCT file has been updated. The MIT tape now includes all 5 USL files: U00U301B, U01U301B, U02U301B, U03U301B AND U04U301B. It also includes the 3 program files: P00P301B, P02P301B and P040301B. This is in addition to the normal PREP into PUB.SYS, and SL.PUB.SYS.

B. KNOWN PROBLEMS

1. The APLDR provides a memory protect error at %1276 relative. It is caused by jumping to the NOP location at the beginning of a JSB routine instead of (NOP+1) location. This was caused by compiling APLDR source without the M05M301B file.

2. The APLDR produces a memory protect error when the Privileged Interrupt board is required. This problem requires changes to DVR63, contact the factory SE support group.

3. Changes to make the Cross Loader more efficient require fixed length records. These files are not compatible with the DNLDSYS and DNLDUSER, in their present form. A workaround is possible by overriding the Crossloader Output file --

   :FILE FILENAME;REC=60,1,V,BINARY;DEV=DISC

C. MISCELLANEOUS

The N03N301A and N04N301A files are merged into file N00N301A.

ON-LINE DIAGNOSTICS, DATECODE 1531

DIAGNOSTIC	NAME	LEVEL	COMMENTS
DISC FILE-2888A	PD360A	00.00	
CART DISC-7900A	PD361A	00.00	
MAGNETIC TAPE	PD362A	03.00	
TERMINAL DATA	PD363A	02.00	
CARD READER	PD365A	05.00	
LINE PRINTER	PD366A	03.00	FOR 2607/10/14
LINE PRINTER	PD366B	01.01	FOR 2607/13/17/18
TELEPRINTER	PD367A	02.00	
TERMINAL CONTROL	PD368A	01.00	
2640 TERMINAL	PD369A	00.00	
CARD PUNCH	PD370A	00.00	
TERM-2600A	PD371A	00.00	
PAPER TAPE READER	PD372A	01.00	
PAPER TAPE PUNCH	PD373A	01.00	
TERM-2762A/B	PD375A	00.01	
CALCOMP PLOTTER	PD376A	00.00	
TERM-2615A	PD378A	01.01	
CARD-READ/PUNCH	PD379A	01.01	

OFFLINE DIAGNOSTICS 30000-1X002

DATE CODE C-L 1630/MAINT 1630

DIAGNOSTIC	NAME	LEVEL	COMMENTS
SDUP	D217A	04.00	
MEMORY PATTERN	PD321B	00.00	
MULTIPLEXOR CHAN	PD322A	00.00	
DISC FILE-2888A	PD323A	01.00	
CART DISC-7900A	PD324A	01.00	
SYSTEM CLOCK	PD325A	00.00	
TELEPRINTER	PD326A	00.00	
FIXED HEAD DISC	PD328A	02.00	
SELECTER CHAN	PD329A	00.00	
TERM-2762A/B	PD330A	01.00	
EXTEND FLT PT	PD331A	00.00	
HSI	PD332A	00.00	
MAGNETIC TAPE	PD333A	01.01	
SSLC INTERFACE	PD334A	01.00	FIX
UI DIAG	PD335A	00.01	
CARD-READ/PUNCH	PD336A	00.01	
DECIMAL FIRMWARE	PD337A	00.00	
SLEUTH	PD211A	02.00	
CART DISC-7905A	PD319A	02.02	

CPU DIAGNOSTICS 30000-1X001

DATE CODE C-L 1403/MAINT 1531

SECTION	NAME	LEVEL
1	PD320A	03.00
2	PD320A1	03.00
3	PD320A2	03.00
4	PD320A3	03.00
5	PD320A4	03.00

SERIES I 1709 SOFTWARE UPDATE

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES I
****

CONTENTS OF M.I.T. DATE CODE 1709

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE	SUPPORT ACCOUNT FILE CONTAINING CHANGES**
*MPE	32000C	00.15	1709	N00N000C.HP32000
*SPL	32100A	06.04	1709	N00N100A.HP32100
*BASIC	32101B	00.06	1709	N00N101B.HP32101
*FORTRAN	32102B	00.07	1709	N00N102B.HP32102
*BASIC COMPILER	32103B	00.04	1709	N00N103B.HP32103
*RPG	32104A	03.04	1709	N00N104A.HP32104
BUILDINT	32150A	03.01	1623	
*EDITOR	32201A	06.01	1709	N00N201A.HP32201
STAR	32204A	01.00	1603	
SCIENTIFIC LIBRARY	32205A	00.02	1430	
*DEL/3000 (**NEW**)	32206A	01.03	1709	N00N206A.HP32206
SDM	32210A	05.00	1508	
COMPILER LIBRARY	32211C	04.05	1701	
*FCOPY	32212A	01.04	1709	N00N212A.HP32212
*COBOL	32213B	02.06	1709	N00N213B.HP32213
*COBOLC	32213C	01.04	1709	N00N213C.HP32213
*SORT/MERGE	32214B	01.04	1709	N00N214B.HP32214
*IMAGE	32215A	04.03	1709	N00N215A.HP32215
*QUERY	32216A	03.04	1709	N00N216A.HP32216
*TRACE	32222A	03.01	1709	N00N222A.HP32222
*XA2100	32223A	01.02	1709	N00N223A.HP32223
XL2100	32226A	02.00	1636	
CALCOMP PLOTTER	30126A	00.01	1640	
2780/3780 EMULATOR	30130B	02.00	1646	
PROG CTRLR/BCS	30300A/ 30361A	00.01	1512	
PROG CTRLR/RTE-C	30301A/ 30361A-1	00.02	1701	
*ONLINE DIAGNOSTICS		-- --	1709	NDONLN.HPONLN
*OFFLINE DIAGNOSTICS		-- --	1709	NDOFFLN.HPOFFLN

*UPDATED/CHANGED IN THIS MIT

**CHANGED FILES NOT DOCUMENTED IN THE FOLLOWING PAGES APPEAR IN SERIES II 1709 DOCUMENTATION.

MPE 32000C.00.15

DATE CODE 1709, N00N000C.HP32000.SUPPORT

I. MPE 32000C.00.15

A. MODULES MODIFIED C.00.XX

MODULE		CHANGE HISTORY													
NAME	NO	C.00.XX													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
INITIAL	0	X		X	X			X	X	X	X	X	X	X		X
SYSDUMP	1	X	X	X			X		X	X			X		X	X
SEGPROC	2	X	X				X			X	X					
SEGDVR	3												X			
DISPATCH	4			X			X				X	X	X			
LOAD	5		X										X			
MAPP	6					X										
UCOP	7	X														
DEVREC	8															
PROGEN	9	X							X	X	X		X			X
ININ	10					X		X			X	X		X		
EXIN	11	X	X	X		X	X	X		X		X			X	X
LOG	12	X									X					
IOPTRDO	13															
IOPTPNO	14						X		X							X
IOPLOTO	15															
IOMDISKO	16			X				X	X	X						
IOFDISKO	17			X				X	X							X
IOTAPEO	18				X				X							
IOLPRTO	19												X	X		
IOCDRDO	20		X				X									
IOCLTTYO	21															
IOTERMO	22											X				
IOCDPNO	23															
IOPRPNO	24					N	X					X	X			X
IOREMO	25															
IOBSCO	26															
IOMDISK1	27	X		N				X	X	X	X					X
PFAIL	30			X	X	X										
FILESYS	50	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
COMM*INT	51	X		X			X					X	X		X	X
STORE/RESTORE	52			X	X			X	X	X			X			X
DIRC	53									X				X		
ALLOCATE	54		X		X				X			X	X			X
DISKSPC	55	X														
MMCORER	56						X		X			X				
MMDISKR	57											X			X	

ABORTRAP	58					X	X		X		X		X				
MESSAGE	59						X		X	X	X						X
CROUTINE	60			X	X					X	X						
IOUTILITY	61	X		X	X			X	X	X						X	X
TTYINT	62		X	X	X			X		X							X
PCREATE	63	X									X						
MORGUE	64			X						X	X						
PROCMAIL	65																
PINT	66						X		X	X			X				X
DATASEG	67	X								X							
IOPM	68		X			X						X	X				
CHECKER	69																
UTILITY	70	X	X	X		X					X				X	X	
SEGUTIL	71	X		X				X		X				X		X	X
LOADER1	72		X	X					X		X					X	
RINS	73						X							X			
JOBTABLE	74	X															
DEBUG	75	X															
NURSERY	76			X													
SYSDPLY	77						X							X			
FIRMWARESIM	78	X								X	X	X					
SPOOLING	79			X	X			X	X	X	X	X			X		X
SPOOLCOMS	80	X					X		X		X			X			X
MESSAGE CAT	--			X			X	X	X	X	X			X	X		

B. ENHANCEMENTS

1. The Initiator will always write the system load map to the disc file LOADMAP.PUB.SYS on system startups from magnetic tape. On a WARMSTART or COOLSTART it is not written to the disc.

2. The user command :DEBUG has been added to the system. See the documentation section for further information.

C. CORRECTIVE SOFTWARE CHANGES

1. The terminal driver in INITIAL was modified to prevent loss of information at 2400 baud during system startup.

2. The headbase sector for the 7905 in INITIAL has been changed to correspond to that of the Series II.

3. INITIAL has been corrected to properly handle suspended jobs during a WARMSTART.

4. SYSDUMP was modified to properly handle a logical device number of zero.

5. PROGEN changed to allow console operator commands to have lower case parameters.

6. EXIN changed to provide for better resolution of the system clock.

7. The paper tape punch driver(IOPTPN0) now properly handles a length of zero.

8. The fixed head disc driver(IOFDISK0) now has a proper CRC check.

9. The hollerith conversion in the card reader/punch drive (IOPRPN0) was corrected.

10. A timing delay was placed in IOPRPN0 to prevent undue degradation to terminal response.

11. The 7905/7920 disc driver(IOMDISK1) was modified to release the disc controller in a similiar fashion as in MPE II.

12. The FOPEN intrinsic can now handle a formsg parameter that begins on an odd byte boundary.

13. FILESYS was modified to insure that the file space calulation never goes negative.

14. The FLOCK intrinsic now insures that all physical I/O is reinitialized or completed prior to the next I/O operation.

15. The intrinsic FREADLABEL was corrected to properly get the address of the user label for ATTIO.

16. STORE/RESTORE would try to restore Series II files with greater than 16 extents. This would result in either part of the file being restored or a catastrophic error. This problem now has been corrected. See documentation section for the resulting new error message.

17. STORE/RESTORE will now list all files that the user tries to store without read access.

18. The initialization of an internal variable was changed in ALLOCATE.

19. A correction was made to MESSAGE to insure that WARN @ and TELL @ messages reach all terminals.

20. IOUTILITY was changed to properly place the physical I/O status word in the IOQ element.

21. TTYINT previously destroyed random core locations when the internal procedure Enablebreak was passed a logical device number that was not a terminal.

22. IOUTILITY and TTYINT were changed such that all FILESYS word reads will be rounded up in cases where an odd number of bytes are read. Previously they were rounded down.

23. PINT was modified to insure that a user could not cause a system pause by setting extra bits in the susp parameter of the intrinsic ACTIVATE.

24. The intrinsic EXPANDUSLF was modified to properly position $OLDPASS.

25. Changed ASL's to LSL's in SPOOLERS and SPOOLCOMS to provied for proper converion to byte addresses.

D. DOCUMENTATION CHANGES

1. A tape file cannot be closed with no rewind. This situation was previously carried as a known problem in MPE. Forthcoming manual revisions will reflect this situation.

2. Chained SIO's on magnetic tape do not work for blocks greater than 4096 words unless the requested length is identical to the block size. This situation was previously carried as a as a known problem in MPE. Forthcoming manual revisions will reflect this situation.

3. When the maximum number of open spoofles is not sufficient to handle all spooling requirements, spooled jobs may cause endless numbers of null list files to be generated. This situation can be overcome by increasing the maximum number of open spoofles. The recommended value is 20, however, a more exact number can be found by examining the usage of your system. Each initial allocation of a spooled device uses one spoofle. A session's single access to a spooled line printer requires one open spoofle; a spooled job requires at least two, one for $STDIN and one for $STDLIST. Each additional access to a file of device class LP requires an additional open spoofle.
One indication that the limit is being reached is allocation failures for spooled devices.
This situaion was previously carried as a known problem in MPE. The forthcoming manual revisions will reflect this situation.

4. The new user command :DEBUG is an extension of the MPE Debug Facility and is used primarily by system programmers. The Debug Facility allows programmers to establish breakpoints within programs from which it is called, and to display and modify data within the stack or registers used by those programs. Because the :DEBUG command effectively invokes the Debug Facility from the MPE Command Interpreter program,

it can establish breakpoints within that program and allow users to operate on the Command Interpreter's stack. The directives and various Debug messages are discussed in the MPE Reference Manual.

5. If the user tries to Restore a Series II file with more than 16 extents the following file rejection message will appear:

GREATER THAN 16 EXTENTS

E. KNOWN PROBLEMS

1. The user commands LISTACCT, LISTGROUP, and LISTUSER can lock the directory indefinitely if the output is written to an unspooled device that is not ready.

2. Lower case :EOD is not recognized as an end-of-file on data accepting devices.

3. Issuing a :DEALLOCATE command for a non existent program file returns an error 217. The error should be ERR 217,52 The 52 is the file system error number returned from FCHECK.

4. The line printers 2613A, 2617A, and 2618A may intermittently report a unit failure condition to the I/O driver that will abort the print operation. This condition has been observed when the unit is brought online after being placed offline while printing.

5. The directory may indicate a table overflow even though there is room available. This situation has been observed when doing a full RELOAD on a system with a full directory, and on systems where large numbers of files are created and purged daily.

6. The EOF on a disc file can exceed the file limit. This situation occurs as a result of files being allocated on sector boundary.

7. The CPU times displayed via a REPORT command periodically far exceeds reality.

8. The command SAVE produces an erroneous message.

II. SUPPORTED UTILITIES

A. UTILITIES MODIFIED C.00.XX

UTILITY	CHANGE HISTORY										
	5	6	7	8	9	10	11	12	13	14	15
DISKEDIT		*									
DPAN	*							X			
FREE		*									
LISTDIR		*			X						
LISTEQ	*							X			
LISTLOG	*										
PATCH		*									
RECOVER	*	X									
SAEDIT	*				X						X
SAVIOUR	*				X						X
SLPATCH		*									

* NEW

B. CORRECTIVE SOFTWARE CHANGES

The headbase sector for the 7905 was changed to correspond to the value in MPE II, and the maximum number of sectors for the 7920 was modified in both SAEDIT and SAVIOUR.

C. KNOWN PROBLEMS

When DPAN finds that the PCB table has been filled, it prints the erroneous message "INVALID FIRST UNASSIGNED ENTRY" AND "INVALID BACKWARD SUBQUEUE POINTER" even though there is no error in the PCB table.

COBOL/3000 HP32213B.02.06

DATE CODE 1709, N00N213B.HP32213.SUPPORT

A. ENHANCEMENTS

1. COPYLIB files can be KSAM files (Series II only).

The following improvements have been added to the Cobol Library in version 4.06:

2. The error message for Error 630 has been changed to ERROR DURING WRITE OF USER LABEL. A File Information Display will also be generated with an Error 630.

3. The error message for Error 631 has been changed to USER LABEL SPACE UNALLOCATED OR ATTEMPT TO WRITE BEYOND LABEL LIMIT.

B. CORRECTIVE SOFTWARE CHANGES

1. Lower case letters in PICTURE clauses were not upshifted.

2. An ON SIZE ERROR clause where the first operand had fewer digits than the result sometimes produced an Error 210 or 211 at compile time.

3. The ACCEPT statement did not work for a table element.

4. A COMPUTE statement at the start of the Procedure Division did not always work properly.

5. The compiler could not recover from an 88-level in a MOVE statement.

6. A GO TO DEPENDING ON statement nested within a conditional clause (such as an AT END clause) produced Error 211 messages--- MULTIPLY OR UNDEFINED INTERAL LABEL.

7. An ADD...TO statement with multiple results of USAGE COMP-3 or DISPLAY left two words on the stack and could produce a stack overflow abort or control flow problems with PERFORM or GOBACK statements.

8. If more than two index names were specified in an INDEXED BY clause, only the first two were handled properly.

9. ADD CORRESPONDING of DISPLAY items aborted at run time.

10. A 4-digit COMP-3 item used as a subscript or compared with an index-name did not work properly.

11. A VALUE clause at the group level in the LINKAGE SECTION was not handled properly. A warning message was not issued and bad object code was generated.

12. Subprograms sometimes aborted with a bounds violation in the initialization procedure.

13. An ADD statement with a numeric edited or alphanumeric item somewhere other than after the word GIVING was not detected as an error.

14. A READ statement without a file name caused the compiler to abort.

15. A GO TO DEPENDING ON statement occasionally aborted at run time.

16. An attempt to perform an empty section generated an Error 201. The error message has been improved to show which section was being performed and where the perform statement is located.

17. Continuation of a word did not work if the line being continued contained a completed non-numeric literal.

18. Text on a line following the period of a COPY statement caused the compiler to abort with a bounds violation or a write error on the list file.

19. Subprograms with large linkage sections sometimes aborted during the initialization procedure.

20. Using an index-name with the wrong table was not detected as an error.

21. A non-unique reference generated an Error 21 for each possible qualifier. Now only one error will be generated.

22. A GOBACK statement in a subprogram would not work if extra words had been left on the stack by some statement in the subprogram.

23. An add of 2 COMPUTATIONAL items each with less than 5 digits and both with the same number of digits after the decimal point would yield incorrect result.

The following problems have been corrected in the COBOL Library in version 4.06:

24. An end-of-file condition was not detected on an ACCEPT statement.

25. After a USE procedure was executed for a read error, the AT END branch was taken.

C. KNOWN PROBLEMS

A SEARCH statement with multiple WHEN conditions sometimes leaves extra words on the stack which can cause either a stack overflow abort or control flow problems with a PERFORM statement. This problem will no longer cause problems with a GOBACK statement in a subprogram (See No. 22, above.)

RELEASE ISSUE OF THE SERIES I ONLINE DIAGNOSTICS.

DATE CODE 1709, NDONLN.HPONLN.SUPPORT

I. MAGNETIC TAPES ASSOCIATED WITH HPONLN

SOURCE	30000-1X011
MAINTENANCE	30000-1X008

II. ON-LINE DIAGNOSTICS, DATE CODE 1531

DIAGNOSTIC NAME	NAME	LEVEL	COMMENTS
DISC FILE-2888A	PD360A	00.00	
CART DISC-7900A	PD361A	00.00	
MAGNETIC TAPE	PD362A	03.00	
TERMINAL DATA	PD363A	02.00	
CARD READER	PD365A	05.00	
LINE PRINTER	PD366A	03.00	FOR 2607/10/14
LINE PRINTER	PD366B	01.01	FOR 2607/13/17/18
TELEPRINTER	PD367A	02.00	
TERMINAL CONTROL	PD368A	01.00	
2640 TERMINAL	PD369A	00.00	
CARD PUNCH	PD370A	00.00	
TERM-2600A	PD371A	00.00	
PAPER TAPE READER	PD372A	01.00	
PAPER TAPE PNCH	PD373A	01.00	
TERM-2762A/B	PD375A	00.01	
CALCOMP PLOTTER	PD376A	00.00	
TERM-2615A	PD378A	01.01	
CARD-READ/PUNCH	PD379A	01.01	

RELEASE ISSUE OF THE SERIES I STANDALONE DIAGNOSTICS.

DATE CODE 1709, NDOFFLN.HPOFFLN.SUPPORT

I. MAGNETIC TAPES ASSOCIATED WITH HPOFFLN

SOURCE	30000-1X005
MAINTENANCE	30000-1X006
CPU COLD LOAD	30000-1X001
NON-CPU C/L	30000-1X002

II. OFFLINE DIAGNOSTICS 30000-1X002, DATE CODE CL 1630/MAINT 1630

DIAGNOSTIC NAME	NAME	LEVEL	COMMENTS
SDUP	D217A	04.00	
MEMORY PATTERN	PD321B	00.00	
MULTIPLEXOR CHAN	PD322A	00.00	
DISC FILE-2888A	PD323A	01.00	
CART DISC-7900A	PD324A	01.00	
SYSTEM CLOCK	PD325A	00.00	
TELEPRINTER	PD326A	00.00	
FIXED HEAD DISC	PD328A	02.00	
SELECTER CHAN	PD329A	00.00	
TERM-2762A/B	PD330A	01.00	
EXTEND FLT PT	PD331A	00.00	
HSI	PD332A	00.00	
MAGNETIC TAPE	PD333A	01.01	
SSLC INTERFACE	PD334A	01.00	
UI DIAG	PD335A	00.01	
CARD-READ/PUNCH	PD336A	00.01	
DECIMAL FIRMWARE	PD337A	00.00	
SLEUTH	PD211A	02.00	
CART DISC-7905A	PD319A	02.02	

III. CPU DIAGNOSTICS 30000-1X001, DATE CODE CL 1403/MAINT 1531

SECTION	FILE NAME	REV	COMMENTS
1	PD320A	03.00	
2	PD320A1	03.00	
3	PD320A2	03.00	
4	PD320A3	03.00	
5	PD320A4	03.00	

IV. SLEUTH

A. VERIFIERS

	VERIFIER NAME	NAME	LEVEL	DATE CODE
*	7920 VERIFIER	VERI7920	00.00	1709

* NEW

B. 7920 VERIFIER PROGRAM

This is a new program used to verify the 7920 disc. It must be writtern to a magtape using 'FCOPY', and then batched in by 'SLEUTH' with the 'BA E' command.

```
:HELLO FIELD.SUPPORT,HPOFFLN
:FILE T;DEV=TAPE;REC=-72,1,F,ASCII
:RUN FCOPY.PUB.SYS
>FROM=VERI7920;TO=*T
>EXIT
(COLD LOAD SLEUTH)
(MOUNT THE VERI7920 TAPE WRITTERN BY FCOPY)
BA E
>10 BA E
>670 10 DEV 0,<DPT>,15,99,<UNIT #>
>11 LET W=1     (IF UNIT TEST DESIRED ELSE W=0)
>12 LET X=1     (IF PACK FORMAT DESIRED ELSE X=0)
>13 LET Y=1     (IF PACK CERTIFICATION DESIRED ELSE Y=0)
>14 LET Z=1     (IF LONG PASS OF PACK CERIFICATION
                   DESIRED ELSE Z=0)
>670 AUTO
>670 RUN
```

Note: to run format, pack cert., or long pass; w must be 0

For additional instructions see manual:
32210-90001 SLEUTH VERIFIER PROGRAM.

SERIES II 1701 SOFTWARE UPDATE

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES II
****

CONTENTS OF M.I.T. DATE CODE 1701

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE	SUPPORT ACCOUNT FILE CONTAINING CHANGES
*MPE	32002A	01.00	1701	N00N002A.HP32002
SPL	32100A	06.02	1624	
BASIC	32101B	00.05	1646	
*FORTRAN	32102B	00.06	1701	N00N102B.HP32102
BASIC COMPILER	32103B	00.02	1646	
*RPG	32104A	03.02	1701	N00N104A.HP32104
*APL/3000	32105A	00.02	1701	N00N105A.HP32105
BUILDINT	32150A	03.01	1619	
*EDITOR	32201A	06.00	1701	N00N201A.HP32201
SCIENTIFIC LIBRARY	32205B	00.01	1634	
*DEL/3000	32206A	01.02	1701	N00N206A.HP32206
*KSAM/3000	32208A	00.03	1701	N00N208A.HP32208
*COMPILER LIBRARY	32211D	00.05	1701	N00N211D.HP32211
*FCOPY	32212A	01.03	1701	N00N212A.HP32212
*COBOL	32213C	01.02	1701	N00N213C.HP32213
*SORT/MERGE	32214B	01.03	1701	N00N214B.HP32214
*IMAGE	32215A	04.02	1701	N00N215A.HP32215
*QUERY	32216A	03.03	1701	N00N216A.HP32216
TRACE	32222A	03.00	1610	
*XA2100	32223A	01.01	1701	N00N223A.HP32223
XL2100	32226A	02.00	1636	
PROG CONTROLLER 30300B/30361B-BCS	30361B	00.00	1621	
*PROG CONTROLLER 30301B/30361B-1-RTE	30361B-1	00.02	1701	N00N301B.HP30301
RJE 2780/3780	30130C	00.02	1646	
CALCOMP PLOTTER	30126A	00.01	1640	
*DIAGNOSTICS	32230A	-- --	1701	N00N230A.HP32230

*UPDATED/CHANGED IN THIS MIT

MPE HP32002A.01.00

DATE CODE 1701, N00N002A.HP32002.SUPPORT

I. MPE 32002A.01.00

--
NOTE - MPE has been updated for this MIT. The file system has been resequenced. If you have source tapes, you should order new ones. References to MPE 00.06 are the same (from this document's point of view) as referring to MPE 01.00.
--

A. MODULES MODIFIED A.00.XX

MODULE		CHANGE HISTORY													
NAME	NO	A.00.XX													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
INITIAL	0		X	X	X		X								
SYSDUMP	1		X	X	X	X	X								
SEGPROC	2		X				X								
SEGDVR	3		X												
DISPATCH	4		X												
LOAD	5		X												
UCOP	7		X		X										
DEVREC	8				X										
PROGEN	9		X	X	X	X	X								
ININ	10				X										
MEMLOGP	11		X		X	X	X								
LOG	12				X		X								
IOPTRD0	13		X	X			X								
IOPTPN0	14		X	X	X	X									
IOPLOT0	15														
IOMDISC0	16														
IOFDISC0	17														
IOTAPE0	18					X									
IOLPRT0	19		X		X										
IOCDRD0	20					X									
IOTERM0	22		X	X	X	X	X								
IOPRPN0	24														
IOREM0	25														
IOMDISC1	27						X								
PFAIL	30				X	X									
FILESYS	50		X	X	X	X	X								
COMM'INT	51			X	X	X	X								
STORE/RESTORE	52						X								
DIRC	53														

ALLOCATE	54					X	X	
DISCSPC	55							
MMCORER	56				X		X	
MMDISKR	57		X	X	X		X	
ABORTRAP	58		X			X		
MESSAGE	59				X		X	
CROUTINE	60						X	
CLOCKIO	61					X		
NRIO	62		X	X		X	X	
PCREATE	63		X			X		
MORGUE	64						X	
PROCMAIL	65							
PINT	66		X			X	X	
DATASEG	67		X		X		X	
CRIO	68		X	X	X	X	X	
CHECKER	69							
UTILITY	70				X	X	X	
SEGUTIL	71		X				X	
LOADER1	72		X					
RINS	73	X		X				
JOBTABLE	74						X	
DEBUG	75				X			
NURSERY	76				X			
STKDUMP	77							
FIRMWARESIM	78							
SPOOLING	79					X	X	
SPOOLCOMS	80					X	X	
MESSAGE CAT	--							

B. ENHANCEMENTS

INITIAL was enhanced such that warmstart can handle spoolfiles which have the squeeze bit set.

C. CORRECTIVE SOFTWARE CHANGES

1. INITIAL was modified to increase the size of the MEMLOGP stack size to avoid stack overflow on FOPEN of memory logging file. INITIAL was also modified to skip user files on parity errors.

2. A fix was made to FILESYS in the 1646 MIT to disallow no-wait I/O in conjunction with multi-access. This capability was required by RJE,therefore it has been restored. Further fixes are necessary for no-wait I/O and multi-access to work for the general case. It will, however, work for one user new.

3. IOMDISCI was modified to work with the new microcode for the 7905 disc controller. INITIAL was modified for the same reason.

4. CRIO was modified to extract the PIN from IOQ before waking the caller. This was necessary to avoid misinterpreting some ATTACHIO changes.

5. LOG was prepped without PM capability so it could not be run as a process.

6. MEMLOGP was modified to avoid wasting disc space when creating the MEMLOG file.

7. IOPTRDO was modified to store a count in DITP+9 instead of DITP+30. Under certain conditions a SF250 could occur.

8. The PRINTOPREPLY procedure was modified to output properly when "large" stacks are used. The routine was using arithmetic shifts instead of logical shifts.

9. SPOOLCOMS was modified to eliminate an SF19 which occurred when trying to use a mag tape configured for output spooling. DELETE and ABORTJOB were modified at the same time to accept lower case "s" and "j".

10. ALLOCATE had to be modified avoid a forms request being issued from SYSDUMP when the FILE equation specified the logical unit number instead of a name.

11. CRIO was modified such that XON will not restart a write if waiting for a write interrupt. This situation lost an interrupt. TIP now runs with interrupts disabled which will prevent two copies of the interrupt handler from running at once on the same unit or DRT.

12. SEGUTIL was modified to adjust the EOF pointer while copying a $NEWPASS type of USL during a USL expansion operation.

13. The LOG process was modified to avoid potential problems when modifying the file label of a log file.

14. MEMLOGP was modified to FUNLOCK the MEMLOG file when the process is abnormally terminated.

15. The FILESYS module has been modified to correct a problem introduced in the last MIT which didn't allow one to FOPEN a mag tape file for append access.

16. FREADLABEL/FWRITELABEL was corrected to use the file's extent map. This should resolve some of the sporadic clobbering of FILE LABELS that have been reported. (See miscellaneous)

17. You are now allowed to access up to 255 extra data segments per process.

18. Pressing BREAK during LOGON with the new :(COMMAND) will now work properly.

19. IOTERM0/NRIO/CRIO have had several changes:

    a. The XOFF is no longer sent for TERM TYPES equal to or greater than 7.

    b. A new TERM TYPE (13) has been added for communicating with the TERMINET.

    c. A new meaning has been added to XON and XOFF. Striking XOFF during a write causes the terminal driver to suspend the output of characters until XON is input.

    d. The current information in the PTAPE BUFFER is output now, even if it's null.

20. A check has been included in CROUTINE to ignore the CONTROLY signal, if the procedure is no longer there.

21. An enable of interrupts, along with the reference counter shift it permitted, was removed from MMCORER. This is to resolve the very rare SF130 problem.

22. A problem in STORE'RESTORE that resulted in an 'E' being printed as the first character of a file name, has been resolved.

23. To resolve an incompatibility between the CX and Series II LOG FILES, the following changes have been made to the Series II.

    a. The last byte of the second reserved word in the Job Initiation Record now contains an ASCII A, B, C, D or E indicating the LOG ON QUEUE.

    b. The last BYTE of the last word in the Spoolfile Done Record now reflects the disposition of the Spoolfile as was done on the CX.

24. A change was installed in IOTAPE0 to handle EOFCHECK properly. This resolves a problem introduced in the 05 MIT which is uncovered when input spooling a mag tape with :DATA and :EOD records. The symptom was that alternate data sets were recognized.

D. DOCUMENTATION CHANGES

The SYSTEM MANAGER/SYSTEM SUPERVISOR manual is being modified to reflect some changes in SYSDUMP/INITIAL. A new question is asked when adding I/O devices if the type is 16. The question is "SPEED IN CHARACTERS PER SECOND". Valid responses are a 0 or carriage return which specify an unknown speed, otherwise specify the speed at which the terminal will be operated(ex: 240,120,60,30,15,14,10).

E. KNOWN PROBLEMS

1. SYSTEM FAILURE TYPE

   a. SF366 - We have had some recently reported spooler problems that cause a SF366.

   b. SF127 - The 127 problem is still with us under very heavy disc loads. We have not received any customer reports on this problem yet, have been able to create it internally. We feel that the solution to this problem will be available on the next MIT.

2. FUNCTIONAL FAILURE TYPE

   a. A session with an outstanding READ cannot be aborted until reception of a carriage return.

   b. The FILESYSTEM seems to be building strange looking files for record sizes between 16384 and 32761 words.

   c. Saving a large file with SADUTIL doesn't seem to work.

   d. When logical records, plus overhead (byte count and EOBLOCK word) fit the computed block size exactly, the last record is written into the next block, thus wasting some space in the current block.

   e. If $STDLIST is passed to FOPEN as the formal designator and is not overridden by the FOPTIONS, the file system will open $STDIN instead.

   f. A session logging on with HIPRI can be logged off under certain conditions, you shouldn't be able to do this.

g. Executing a FORTRAN program using the command intrinsic with no CR at the end causes a SF311.

h. A call to XARITRAP with MASK=0 does not return CCG to indicate traps disabled.

i. A potential problem exists in SYSDUMP. If you are adding a driver, you may run out of BANK 0. To avoid this problem reduce the system buffers.

F. MISCELLANEOUS

1. For some time now, members of the MPE team have requested an increase in the size of the LDT (Logical Device Table), so new features may be added to the system. MPE will have a logical extension to the LDT.

   The extension will be a new table called LDTX (Logical Device Table eXtension). The LDTX will be located in the same DST as the LDT and Device Class Table (DST 14(10)). The LDTX will follow the Device Class Table.

   The first enhancement to take advantage of the LDTX is handling of terminal after FOPEN. A question was added to the I/O configuration dialog to ask terminal speed (characters/second) thus making it possible (with minor changes in other MPE modules, SMOP) to allocate terminals without having to know the speed or type (the terminal must be configured correctly).

   There is one possible drawback in adding this table. Since this table contains configuration information, it is necessary that it be put in the table by SYSDUMP and on the Disc by INITIAL. This means special tables for INITIAL. INITIAL can handle the new table complete if an "UPDATE" is executed at cold load time when going to 01 from an older version. But it is recommended that once updated to 01, do not go backwards to an older system unless absolutely necessary. Since LDTX is a new table,. every time a backwards update is done disc space is lost. If done enough times without recovering lost disc space, this could result in running out of disc space and require a reload. It does seem, however, that a backwards UPDATE will work. There may be overlooked problems with a backwards UPDATE, therefore it is not recommended.

2. IF your system was one of the few that had problems with FILE LABELS being destroyed, fix C.16. should resolve it, however it may cause a problem of a

different type. The resolved problem occured when the USER LABELS were written beyond the first extent. Since both STORE and SYSDUMP have always used the extent map as it was intended, they did not save any USER LABELS past the first extent. Therefore the possibility of losing the label was high. There is no simple way of recovering from this situation especially if the USER LABEL has been destroyed. The best thing to do is to read the labels of suspected files with FREADLABEL and save them in a unique file (prior to updating), then rewrite them with FWRITELABEL after updating to 01.00.

3. At the end of a SYSDUMP a warning message will be generated indicating not all of the system files have been dumped. As long as the system file names are MPMON, IODS0, IOREM0, IOPLT0, and CSHBSC0 everything is still ok.

II. SUPPORTED UTILITIES

A. UTILITIES MODIFIED A.00.XX

UTILITY	CHANGE HISTORY A.00.XX													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DISKEDT2														
DPAN2		X	X	X	X									
FREE2														
LISTDIR2														
LISTEQ2														
LISTLOG2		X	X			X								
PATCH2														
MEMLOGAN														
MEMTIME														
SADUTIL														
SLPATCH														
SPOOK						X								

B. CORRECTIVE SOFTWARE CHANGES

LISTLOG2 was cleaned up to avoid outputting extra blank pages. It has also been corrected to run in BATCH.

FORTRAN HP32102B.00.06

DATE CODE 1701, N00N102B.HP32102.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

1. When the FORTRAN list file has a record length shorter than the length of the line the compiler needs to print, the compiler must fold the line into two or more lines. Two problems related to this folding were corrected. First, if the logical record length of the list file is an odd number of bytes, or if $CONTROL LOCATION is enabled and the list file record length is even, the compiler tended to truncate the last character. Secondly, if the list record fit exactly one line and $CONTROL LOCATION was enabled, the compiler forgot about the carriage control character and truncated the last character in the list line.

2. Use of $CONTROL LOCATION,MAP on a list file with a line width between 40 and 93 characters caused a write failure to the FORTRAN scratch list file FTNUT5 when a page eject /new page heading must be printed in the middle of the variable map. (BR 1330)

3. When using $CONTROL CROSSREF, the compiler would abort during the listing of the cross reference if the program unit contained an integer simple variable unit number which had not been previously seen by the compiler. i.e.,

```
$CONTROL CROSSREF
      WRITE(IOUT,*)
      STOP
      END
```

4. Using DO loops with INTEGER*4 limits in subroutines or functions which require many distinct local variables may result in accidental destruction of other local data or in an bounds violation. Under the rules of FORTRAN, the user is not allowed to modify the limit of a DO loop while that DO loop is executed. To achieve this, a dummy local variable is generated to hold the limit. When many

local variables are required, the compiler uses rather unusual code to get around the addressing limitations of the hardware. The conjunction of these two unusual conditions with double integer DO loop limits resulted in the index register being twice the value required when accessing the temporary DO loop limit. Consequently, seemingly unrelated memory locations were destroyed.

B. KNOWN PROBLEMS

The use of dynamic array bounds in a program unit which contains secondary entry points is legal only if the dynamic bounds are passed in each entry point, even if the code being executed does not access the array. This is because the space must be allocated for the array even if it is not used. The compiler does not now check to make sure that the bound is indeed passed at each entry point. When entry is made at one of the points which lacks the dynamic bound variable, the user's program will abort with a bounds violation in the program unit initial- ization code. At some point the compiler will be changed to detect this user error.

RPG SUBSYSTEM HP32104A.03.02

DATE CODE 1701, N00N104A.HP32104.SUPPORT

A. ENHANCEMENTS

1. If an edit word begins with a constant, that constant will always appear, before it was blanked out.

2. Cross reference is now single spaced.

3. A 1 in column 22 of the header record will cause all direct file accessed to begin with one. (RPG will subtract one from any record number given).

B. CORRECTIVE SOFTWARE CHANGES

1. Z-ADD to an array with more than 255 elements did not modify all the elements.

2. Programs with data space larger than 32k bytes had bounds violations.

3. A SNULL RSAM file was not handled properly.

4. An edit code of C used with a field with no integers (decimals only) resulted in an incorrect edit.

5. A Z-SUB to an array zeroed out the array.

6. An * as an indicator repeat did not always work.

7. If RSAM and KSAM were both on the system an output RSAM file had its data deleted, including control information.

8. A bounds violation would occur sometimes if an array were loaded through the input specifications.

9. An array with two different subscripts in the same calculation could produce erroneous results.

10. Input record sequence numbers could have caused a bounds violation.

11. If more than 7 OR lines appear in Calculations a terminal error is given.

12. If more than 20 AND/OR lines appear for one output record a terminal error is given.

13. If there are many AND/OR records a compile time error or prep time error could have occurred.

14. INnn on an RLABL resulted in a terminal error at compile time.

15. Bounds violation would sometimes occur while compiling an RPG program with input sequencing.

APL HP32105A.00.02

DATE CODE 1701, N00N105A.HP32105.SUPPORT

MISCELLANEOUS

1. APL requires in addition to the program file a set of PROMs mounted on the EIS board; this contains the extra instructions which APL executes. Without these instructions an unimplemented instruction error will result.

2. The update level for this version is 0 and the fix level is 02. There are no known bugs to this version.

EDITOR HP32201A.06.00

DATE CODE 1701, N00N201A.HP32201.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

1. Attempting to TEXT a file created by a KEEPQ command, or a K..... file, which was not created by the log-on user, will result in an error 94, CREATOR CONFLICT. Previously such files were either not closed, or purged. With this fix files are closed with a disposition of no change, so that such files remain intact, and are not purged.

2. Text files will now be opened initially with these AOPTIONS:

   non no-wait (4:1)=0
   non multi-access (6:1)=0
   nobuf (7:1)=1 (to permit EDITOR deblocking, for speedier I/O)
   default access (8:2)=0 (permits reading across groups and accounts)
   no dynamic locking (10:1)=0
   multirecord access (11:1)=1 (to permit EDITOR deblocking)
   read only access type (12:4)=0

   The purpose of this is to permit access to files across accounts and groups, where security provisions permit. Formerly the ability to access across accounts and groups was also limited by the locking security provision, as implied by the exclusive access read option, (8:2)=2. Now security restrictions are explicit, so that when a group or account specifies read access to anyone, locking access will not be necessary in order to TEXT the file in the EDITOR.

3. LENGTH, RIGHT, and LEFT values are now retained with the work file, and files kept using a K[EEP]Q command. The effect of this change is that whenever such files are made into work files using the TEXT command the LEFT, RIGHT, and LENGTH values are restored to their previous actual value, rather than to a default setting.

4. BEGINQ or BQ and BEGIN or B with QUIET set now have the same effect.

5. The method for computing the actual work file size has been corrected so that extremely large work files no longer require the use of the SET SIZE command. Some users wish to edit files of approximately twenty thousand (20,000) records. Files beyond four thousand (4,000) records still will be provided with an additional work space of two thousand (2,000) records as documented.

6. A warning message has been added to the KEEP command.

7. The prompting to enter a string to the Z::= command will now occur when the EDITOR is used interactively and when batch is not set.

8. GATHER ALL with a very large work file (over 18,000 lines) will no longer cause a file error.

9. Under special circumstances, not all occurrences of a string were found by the CHANGE command. This had occurred when only one line was to be changed, and the length of the replacement string was greater than the length of the string replaced.

10. Doing a TEXT on a file with over 172 but less than 256 bytes per record had caused a spurious warning message.

11. XPLAIN JOIN and XPLAIN J now give the same results.

B. KNOWN PROBLEMS

1. Use of the CONTROL Y during a MODIFY is ambiguous: it may be interpreted to restart the MODIFY using the original line value; also, it may be interpreted to exit from the command.

2. Re-setting the SIZE value may inhibit texting a file in its entirety.

3. Under special circumstances, a request to INSERT between lines previously created by INSERT can cause a file error.

4. If an INSERT runs out of line numbers between two preexisting lines, the character string being carried forward by INSERT is lost.

5. The value of LAST as a column position in a CHANGE command is derived from the current line BEFORE the CHANGE command was entered, rather than the line to which the CHANGE command is directed.

6. When Z:: is active, an "&" is interpreted as an END command.

7. XPLAIN ALL from a job will cause a file error.

DEL/3000 HP32206A.01.02 (FORMAINT)
HP32206A.01.03 (LIBRARY PROCEDURES)

DATE CODE 1701, N00N206A.HP32206.SUPPORT

CORRECTIVE SOFTWARE CHANGES

1. The procedure GETFORM no longer generates an infinite series of form preamble sequences if the user buffer is less than 74 bytes long.

2. The procedure EDITFIELD will now bypass edit specifications with no edit procedure name.

3. FORMAINT has been altered to allow any characters to be used in forming edit procedure names (including embedded blanks)

4. Word 37 of the DEL communications area is now used by GETFORM to control generation of form preamble escape sequences.

KSAM/3000 HP32208A.00.03

DATE CODE 1701, N00N208A.HP32208.SUPPORT

CORRECTIVE SOFTWARE CHANGES

When keyblocks are higher than 2 levels and use FFINDBYKEY with relop=1 or 2, the current record pointer might not point to the correct place in the file.

COMPILER LIBRARY HP32211D.00.05

DATE CODE 1701, N00N211D.HP32211.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

If one used the TAB operator (Tnnn) in a FORTRAN FORMAT statement such that one tabbed backwards then forwards, but not to the end of the line, all data following the very last item entered was lost.

B. MISCELLANEOUS

This version of the COMPILER LIBRARY only supports four word long floating point operations.

FILE COPIER, HP32212A.1.03

DATE CODE 1701, N00N212A.HP32212.SUPPORT

A. ENHANCEMENTS

1. The previously announced enhancement for HP2644A terminals is for only to the HP3000 Series II because of dependencies upon enhancements to MPE in the file system.

2. Character code conversion, E.G. between EBCDIC and ASCII, is processed through the use of the C'TRANSLATE system intrinsic to provide standardized conversion.

3. With this version of FCOPY, KSAM access will be supported when implemented on a system. Existing users do not need to make any changes in the manner in which they use this product. This enhancement applies only when this product is used on an HP3000 Series II. Should this enhancement be used otherwise execution errors will occur.

4. KSAM use of FCOPY is as follows:

   a. The option "NOKSAM" means that a fromfile which is a KSAM file is not to be treated as such, but rather as a serial file.

   b. The option "KEY=<KEYLOCATION>" means that a KSAM file is to be accessed using the key identified by a byte offset indicated by the value of <KEYLOCATION>.

      1) Permissibile values of <KEYLOCATION> and their use are only greater than zero.

      2) A <KEYLOCATION> value of zero means that the KSAM fromfile is to be accessed in chronological order, that is, in the order in which the data has been entered. Note that this could be different from the physical order in a volatile file.

      3) A positive <KEYLOCATION> value means that the key beginning at the byte offset, as indicated when the KSAM file was created.

   c. The absence of either "NOKSAM" or "KEY=<KEYLOCATION>" when used with a KSAM fromfile means that the file is to be accessed in primary key sequence.

5. The option "NEW" may not be used with KSAM fromfiles.

6. Use of the string or pattern "SUBSET" option is encouraged as an immediate means of retrieval.

B. CORRECTIVE SOFTWARE CHANGES

1. Use of the option SKIPEOF works properly for tape devices.

2. Error message prompting in sessions is consistent with MPE.

3. Use of the IGNERR option now works properly.

4. FROM=$NULL (which makes more sense in a file equation) no longer aborts the program.

5. The program now differentiates properly between the file reference *<FILENAME> and <FILENAME>.

COBOL HP32213C.01.02

DATE CODE 1701, N00N213C.HP32213C.SUPPORT

A. ENHANCEMENTS

1. Main program segments no longer have to begin at PB+0 in their respective segments.

2. File specified in an FD entry can be locked/unlocked with the use of a new parameter on the $CONTROL command and/or a new parameter in the SELECT clause:

   a. $CONTROL LOCKING allows all files to be locked/unlocked. The command may appear anywhere in the source program.

   b. A ,L after the forms msg parameter in the SELECT clause will allow that particular file to be locked/unlocked.

   c. If a $CONTROL LOCKING command is present, it will override any SELECT clauses without the locking parameter.

The following procedures have added to the COBOL Library in VERSION 4.05:

3. COBOLLOCK. This procedure has three parameters and may be called directly from a COBOL program to lock a file. The 2nd and 3rd parameters are both PIC S9(4) USAGE COMP. Parameter 1 is the file-name from the FD entry of the file to be locked. Parameter 2 corresponds to the lockcond parameter of the FLOCK intrinsic and Parameter 3 is used to return the condition code from FLOCK. It will be a -1 (CCL), 0 (CCE), or +1 (CCG).

4. COBOLUNLOCK. This procedure has two parameters and may be called directly from a COBOL program to unlock a file. The 1st parameter is the file-name from the FD entry of the file to be unlocked. The second parameter is a PIC S9(4) USAGE COMP item in which the condition code from FUNLOCK will be returned. The same condition code values as for COBOLLOCK apply.

B. CORRECTIVE SOFTWARE CHANGES

1. Level 77 entries with USAGE clauses were being erroneously detected as errors if the last group item in the File Section had a different USAGE clause.

2. If an Undefined Procedure Name error exceeded the limit of allowable errors, the procedure names would not be listed with the error message.

3. The ASCII/Binary parameter of the SELECT clause did not work properly.

4. The construct IF condition THEN NEXT SENTENCE without an ELSE clause caused an Error 211--Undefined or Multiply Defined Internal Label.

5. An 01 Level REDEFINES clause in the Linkage Section did not always work properly.

6. Moving a table element to an alphanumeric edited item did not work.

7. A divide in a COMPUTE statement in which the divisor was an expression of more than one COMP item did not work properly.

8. A SORT statement nested within a conditional clause (such as an AT END clause) caused an Error 211--Undefined or Multiply Defined Internal Label.

9. A COPY statement did not always find the library.

10. Several problems with decimal point alignment in REMAINDERS were fixed.

11. Non-dynamic subprograms could not be put into an RL.

12. Moving a non-dynamic subprogram from the last segment in a USL file into another segment produced an Invalid Program File.

13. Compiling a subprogram with secondary entry points with names similar to the subprogram name into an existing USL file would not always inactivate previous secondary entry points.

14. Large subprograms with multiple parameters would sometimes cause the compiler to loop or abort.

15. A program with more than 1000 VALUE clauses would not PREP.

16. A PICTURE clause containing floating minus (or plus) signs to the right of the decimal point truncated the last digit.

17. A BLOCK CONTAINS n CHARACTERS clause without a RECORD CONTAINS clause did not compute the proper blocking factor.

18. Decimal point alignment was not performed on a move of a numeric literal item to a numeric item.

19. Moving a value to TALLY in a subprogram did not work.

20. A non-standard level number was not always handled properly. (Example: an 01 level followed by an 05 followed by an 02.)

C. KNOWN PROBLEMS

A READ statement in which the file-name is omitted causes the compiler to abort.

SORT HP32214B.01.03

DATE CODE 1701, N00N214B.HP32214.SUPPORT

A. ENHANCEMENTS

The size of the scratch file used by a stand-alone sort will not be shown in the statistics at the end of the program.

B. CORRECTIVE SOFTWARE CHANGES

Pressing CONTROL-Y during a stand-alone sort before the first record had been read produced an improper status message. The message will now be either 0 records read or initialization phase.

IMAGE/3000 HP 32215A.04.02

DATE CODE 1701, N00N215A.HP32215.SUPPORT

ENHANCEMENTS

BIMAGE (the BASIC - IMAGE interface package) has been added to IMAGE as a supported product. BIMAGE permits IMAGE intrinsics to be called from BASIC programs. See the IMAGE manual (DEC. 1976) for documentation.

QUERY/3000 HP32216A.03.03

DATE CODE 1701, N00N216A.HP32216.SUPPORT

A. ENHANCEMENTS

The VERSION command now prints the current version, update and fix level of BIMAGE (the BASIC-IMAGE interface) in addition to all other IMAGE routines and program files.

B. CORRECTIVE SOFTWARE CHANGES

On data sets containing at least 65,536 records, certain forms of the FIND command could cause QUERY to abort with the following message:

```
ABORT:  QUERY.PUB.SYS.%14.%3275
PROGRAM ERROR #1:  INTEGER OVERFLOW
```

This condition has been corrected.

C. KNOWN PROBLEMS

QUERY does not accept a print position greater than 132 in a REPORT statement, even though the output is directed to a lineprinter configured to 136 characters. A temporary sollution is to issue the file command:

```
FILE OSLIST;REC=-136,1,F,ASCII
```

CROSS ASSEMBLER (XA2100) HP32223A.01.01

DATE CODE 1701, N00N223A.HP32223.SUPPORT

A. ENHANCEMENTS

The sequence number of the previous error has been added to the Pass 2 error messages.

B. CORRECTIVE SOFTWARE CHANGES

1. Floating point numbers were represented incorrectly when the Cross-Assembler was compiled on a Series II system.

2. Equating the listfile to $NULL would cause the Cross Assembler to abort.

3. In the 21MX instructions which require two operands both of which are addresses, if the second operand was of the form "*+n", an incorrect address was generated. The instructions affected are: TBS, SBS, CBS, JRS.

4. If an EQU statement was used to equate a label to a symbol which was declared in an EXT statement, the label name appeared in an EXT record. This resulted in undefined externals at load time.

C. KNOWN PROBLEMS

The title which is specified in the HED statement is sometimes printed incorrectly.

PROGRAMMABLE CONTROLLER HP30301B.00.02

DATE CODE 1701, N00N301B.HP30301.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

This version of the programmable controller is unchanged as far as the program modules are concerned: IOREMO, DNLDUSER, DNLDSYS, LINETEST, XRTCGEN, and the RTE relocatable binary modules. This date code was changed to maintain version compatibility with the SERIES I MIT tape distribution. The version level was changed from A.0.1 to A.0.2. The previous change in substance was MIT 1636. The INSTRUCT file has been updated. The MIT tape now includes all 5 USL files: U00U301B, U01U301B, U02U301B, U03U301B, and U04U301B. It also includes the 3 program files: P00P301B, P02P301B, and P040301B.

B. KNOWN PROBLEMS

1. The APLDR provides a memory protect error at %1276 relative. It is caused by jumping to the NOP location at the beginning of a JSB routine instead of (NOP+1) location. This was caused by compiling APLDR source without the M05M301B file.

2. The APLDR produces a memory protect error when the Privileged Interrupt board is required. This problem is still under investigation.

3. Changes to make the Cross Loader more efficient require fixed length records. These files are not compatible with the DNLDSYS and DNLDUSER, in their present form. A workaround is possible by overriding the Crossloader Output file --

```
:FILE FILENAME;REC=60,1,V,BINARY;DEV=DISC.
```

C. MISCELLANEOUS

1. DNLDUSER was changed in MIT 1636. It was modified to transmit only the significant words in an absolute record, rather than the actual number of words in the record. This was done by using the length in the left 8 bits of the first word and adding three. NOTE: the XL2100 HP32226A.02.00 DATECODE 1636 or subsequent versions must be used.

2. XRTCGEN was changed in MIT 1636. The interrupt table was not being handled correctly when a Power Fail routine was included as any entry but the last one.

   Type 7 utility subroutines were not being appended to each program which called them. They were being loaded only once. This was corrected with the addition of two new commands: LOCKSYS and CLEARPROG. When a type 7 routine is referenced by a program it is append- ed to the program. CLEARPROG then clears out the program and entry points. A subsequent program which references the same routine results in the appending the subroutine again. The change requires the use of XL2100A.02.00 or later. Earlier versions will abort with an invalid command message.

DIAGNOSTICS HP32230A

DATE CODE 1701, N00N230A.HP32230.SUPPORT

I. MAGNETIC TAPES ASSOCIATED WITH HP32230A

Source	32230-1X001
Maintenance	32230-1X002
CPU Coldload	30000-1X016
NON-CPU C/L	30000-1X017

II. MANUALS ASSOCIATED WITH HP32230A

32230-60001
32230-60002

III. CPU DIAGNOSTICS 30000-1X016, DATE CODE 1623

SECTION	NAME	LEVEL
1	PD420A	01.00
2	PD420A1	01.00
3	PD420A2	01.00
4	PD420A3	01.00
5	PD420A4	01.00
6	PD420A5	01.00
7	PD420A6	01.00
8	PD420A7	01.00
9	PD420A8	01.00
10	PD420A9	01.00
11	PD420A10	01.00
12	PD420A11	01.00
13	PD420A12	01.00
14	PD420A13	01.00

IV. STAND-ALONE DIAGNOSTICS 30000-1X017, DATE CODE 1701

A. DIAGNOSTICS CHANGED

DIAGNOSTIC NAME	NAME	LEVEL	OCTAL FILE #
SLEUTH	PD411A	01.00	(01)
SDUPII	PD417A	01.00	
*CART DISC-7905A	PD419A	01.03	(02)
MEMORY PATTERN	PD421A	01.00	(03)
MULTIPLEXOR CHAN	PD422A	01.00	(04)
DISC FILE-2888A	PD423A	01.00	(05)
CART DISC-7900A	PD424A	01.00	(06)
SYSTEM CLOCK	PD425A	01.00	(07)
TERMINAL DATA	PD427A	01.00	(10)

FIXED HEAD DISC	PD428A	01.00	(11)
SELECTER CHAN	PD429A	01.00	(12)
*FAULT CORRECTING MEM.	PD430A	01.01	(13)
EXTENDED INSTRUC SET	PD431A	01.00	(14)
MAGNETIC TAPE	PD433A	01.00	(15)
*SSCL INTERFACE	PD434A	01.01	(16)
UI DIAG	PD435A	01.00	(17)
TERMINAL CONTROL	PD438A	01.00	(20)
CALCOMP PLOTTER	PD439A	01.01	(21)

*UPDATED/CHANGED in this MIT

B. CORRECTIVE SOFTWARE CHANGES

1. STAND ALONE DISC DIAGNOSTIC.

In step 23, the checking for the new and old version of microcode was not the same as listing of error message for the sector address. The bug was fixed.

2. STAND ALONE FAULT CORRECTING MEMORY DIAGNOSTIC.

The problem in PROCEDURE WRIT'READ called by steps 11,12,14 and 15 in section 3 was fixed. The problem appeared when more than 2 full banks were executed.

3. STAND ALONE SSLC INTERFACE DIAGNOSTIC.

STEP 54 was modified to initialize the 30055 board to 2400 baud. This was necessary due to a change of vendor of the 1820-0834 chip (U91 and U136).

V. ONLINE DIAGNOSTICS

DIAGNOSTIC NAME	NAME	LEVEL
CARD READER	PD465A	01.00
LINE PRINTER	PD466A	01.00
2640 TERMINAL	PD469A	01.00
TERM-2762A/B	PD475A	01.00
DISPLAY TERMINAL 2644	PD477A	01.00
TERM-2615A	PD478A	01.00
CARD-READ/PUNCH	PD479A	01.00
OPTICAL MARK READER	PD480A	00.00

SERIES II 1709 SOFTWARE UPDATE

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES II
****

CONTENTS OF M.I.T. DATE CODE 1709

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE	SUPPORT ACCOUNT FILE CONTAINING CHANGES
*MPE	32002A	01.01	1709	N00N002A.HP32002
*SPL	32100A	06.04	1709	N00N100A.HP32100
*BASIC	32101B	00.06	1709	N00N101B.HP32101
*FORTRAN	32102B	00.07	1709	N00N102B.HP32102
*BASIC COMPILER	32103B	00.04	1709	N00N103B.HP32103
*RPG	32104A	03.04	1709	N00N104A.HP32104
*APL/3000	32105A	00.03	1709	N00N105A.HP32105
*DS/3000	32190A	00.00	1709	N00N190A.HP32190
BUILDINT	32150A	03.01	1619	
*EDITOR	32201A	06.01	1709	N00N201A.HP32201
SCIENTIFIC LIBRARY	32205B	00.01	1634	
*DEL/3000	32206A	01.03	1709	N00N206A.HP32206
*KSAM/3000	32208A	00.04	1709	N00N208A.HP32208
COMPILER LIBRARY	32211D	00.05	1701	
*FCOPY	32212A	01.04	1709	N00N212A.HP32212
*COBOL	32213C	01.04	1709	N00N213C.HP32213
*SORT/MERGE	32214B	01.04	1709	N00N214B.HP32214
*IMAGE	32215A	04.03	1709	N00N215A.HP32215
*QUERY	32216A	03.04	1709	N00N216A.HP32216
*TRACE	32222A	03.01	1709	N00N222A.HP32222
*XA2100	32223A	01.02	1709	N00N223A.HP32223
XL2100	32226A	02.00	1636	
PROG CONTROLLER 30300B/30361B-BCS	30361B	00.00	1621	
PROG CONTROLLER 30301B/30361B-1-RTE	30361B-1	00.02	1701	
*RJE 2780/3780	30130C	00.03	1709	N00N130C.HP30130
CALCOMP PLOTTER	30126A	00.01	1640	
*DIAGNOSTICS	32230A	-- --	1709	N00N230A.HP32230

*UPDATED/CHANGED IN THIS MIT

MPE HP32002A.01.01

DATE CODE 1709, N00N002A.HP32002.SUPPORT

I. MPE 32002A.01.01

A. MODULES MODIFIED A.01.XX

MODULE		CHANGE HISTORY													
NAME	NO	A.01.XX													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
INITIAL	0	X													
SYSDUMP	1	X													
SEGPROC	2														
SEGDVR	3														
DISPATCH	4														
LOAD	5														
UCOP	7														
DEVREC	8														
PROGEN	9	X													
ININ	10	X													
MEMLOGP	11														
LOG	12														
IOPTRDO	13														
IOPTPNO	14														
IOPLOTO	15														
IOMDISCO	16														
IOFDISCO	17														
IOTAPEO	18	X													
IOLPRTO	19														
IOCDRDO	20														
IOTERMO	22	X													
IOPRPNO	24														
IOREMO	25														
IOMDISC1	27														
PFAIL	30														
FILESYS	50	X													
COMM'INT	51	X													
STORE/RESTORE	52	X													
DIRC	53														
ALLOCATE	54	X													
DISCSPC	55														
MMCORER	56	X													
MMDISKR	57	X													
ABORTRAP	58														
MESSAGE	59														
CROUTINE	60	X													
CLOCKIO	61	X													
NRIO	62	X													
PCREATE	63														

MORGUE	64	X
PROCMAIL	65	
PINT	66	X
DATASEG	67	X
CRIO	68	X
CHECKER	69	
UTILITY	70	X
SEGUTIL	71	
LOADER1	72	
RINS	73	
JOBTABLE	74	X
DEBUG	75	
NURSERY	76	
STKDUMP	77	
FIRMWARESIM	78	
SPOOLING	79	X
SPOOLCOMS	80	X
MESSAGE CAT	--	

B. ENHANCEMENTS

1. Four new terminal subtypes have been introduced. The new subtypes, 4 through 7,are equivalent respectively to subtypes 0 through 3 with the exception that NO speed sensing will be performed.

   For these new subtypes, the speed will be determined by the speed specified in SYSDUMP's configuration dialogue. INITIAL and SYSDUMP have been modified to insure that a speed is specified in the configuration dialogue for any terminal with a subtype greater than 3.

SUBTYPE	DESCRIPTION
0	Hardwired,speed sensing
1	103 modem,speed sensing
2	202 modem,speed sensing
3	European modem,speed sensing
4	Hardwired,speed specified
5	103 modem,speed specified
6	202 modem,speed specified
7	European modem,speed specified

   These new subtypes are designed to be used in a situation where the terminal or modem may be expected to present a steady BREAK when for instance the power is off. The steady BREAK would otherwise disrupt speed sensing for other devices on the controller.

   CAUTION- once a subtype of 4 through 7 is used, it must be deconfigured to UPDATE with a system prior to MPE 01.01. If this is not done, the system will crash with a SF 201.

C. CORRECTIVE SOFTWARE CHANGES

1. SPOOLING has been modified to correct a problem in streaming a job where S-DB>16K.

2. SPOOLCOMS was modified to return the correct error number to CI if the STREAM failed for some reason.

3. MMCORER was modified to correct a situation where disc IOQ's could get destroyed. This resolves the SF 127 failures that have been reported.

4.a. INITIAL will now output "OUT OF MEMORY" if it runs out of BANK 0 when creating system processes.

b. INITIAL will now restart SUSPENDED jobs after a WARMSTART as it is supposed to do.

5. SYSDUMP has been modified to correct:

a. A problem of not cleaning up the LPDT before I/O configuration changes.

b. A problem of perpetual looping in the "DELETE ADDITIONAL DRIVER" dialog.

c. A problem which caused an ABORT if a carriage return was used as input to the "MESSAGE CATALOG FILE NAME".

d. The default values.

6. COMMAND has been corrected to set ERROR to 0 if no error has occured. Another problem was corrected which wouldn't allow a BASIC program to be executed from a JOB.

7.a. IOTERM0 was corrected to return the correct sense of echo in enable and disable echo FCONTROLS.

b. Parity and lost data errors will now be logged properly. The previous test wouldn't get these errors logged.

8. ININ was modified to correctly execute paired stack ops. Before this modification, the left half of the pair was being re-executed if a stack overflow occured while a paired stack op instruction was being executed.

9.a. A carrier fail counter has been added to CRIO for modem operation. If carrier fails more than 50 times without a successful I/O completion, then a disconnect is requested.

b. CRIO now saves and restores read data parity during write back of first character for term type=11.

10. A bug was corrected in ALLOCATE that was causing SF 366's. This problem occurred most often when the console operator was changing the output fence in order to control SPOOLED output files.

11.a. The file system will now provide an error message(ERR 8) when the record size exceeds 16383 words.

b. $STDINX will now be accepted as a formal designator.

12. NRIO has been corrected to default to odd parity for terminals on initialization of the system as well as new sessions.

13. JOBTABLE has been modified to pick up the LDEV info properly from JMAT. This resolves a problem introduced in the 1701 MIT that would cause a SF 206. JOBTABLE was modified to correct another problem which caused spurious clobbering of JMAT which in turn wouldn't allow a power fail to recover automatically.

14. IOTAPE0 has been modified to dynamically test for BOT and the DIT logical BOT word. If not both then a rewind can occur.

15. CREATE has been modified to make a process CRITICAL if it's delayed on the Timer Request List. This fix prevents loss of the TRL entries if the process is aborted before the full time delay has occured.

16. GETPROCID has been corrected to return the PIN of the first SON if the parameter passed is <= 0 as stated in the manual.

17. PAUSE now returns CCG if a Timer Request List entry is not available. Prior to this fix, a SF 25 would occur.

18. CLOCKIO has been modified to properly report a SF 3 instead of a SF 25 when no TRL entries are available.

19. PROGEN now starts RESTARTable JOBS after a warmstart as described in the manual.

20. MMDISKER has been corrected to alleviate a situation that can cause the overlaying of a segment being used by a process being prepped. This fix should clear up some of the mysterious/unresolved dumps we have seen where memory links have been destroyed,etc.

D. DOCUMENTATION CHANGES

Several minor changes have been input to the SYSTEM MANAGER, SYSTEM SUPERVISOR, and INTRINSIC manuals.

E. KNOWN PROBLEMS

1. SYSTEM FAILURE TYPE

   No known failures of this type.

2. FUNCTIONAL FAILURE TYPE

   a. A session with an outstanding READ cannot be aborted until reception of a carriage return.

   b. Saving a large file with SADUTIL doesn't seem to work.

   c. When logical records, plus overhead (byte count and EOBLOCK word) fit the computed block size exactly, the last record is written into the next block, thus wasting some space in the current block.

   d. A session logging on with HIPRI can be logged off under certain conditions. You shouldn't be able to do this.

   e. Executing a FORTRAN program using the command intrinsic with no CR at the end causes a SF311.

   f. A call to XARITRAP with MASK=0 does not return CCG to indicate traps disabled.

F. MISCELLANEOUS

1. A new system failure(202) has been added in CRIO in an attempt to catch an illegal TBUFF return. One of our customers has reported a problem in this area. If anyone gets this SYSTEM FAILUPE please get a dump and report it immediately.

2. There are two new files in the PUB group of the SYS account. The first is a STREAM job file to help you give us all the information we at HEWLETT-PACKARD need to help you with your MPE problems. The second is a program file which will produce Software Maintenance Request forms for you to report subsystem problems and enhancements.

   The first file (STREAM job) is called "DUMPJOB." It will:

   a. Generate a Software Maintenance Request form.

   b. Obtain a copy of the loadmap from the file "LOADMAP" in the PUB group of the SYS account. This is an absolute "must" item for anyone attempting to analyze a system dump listing.

c. Obtain a copy of your system I/O configuration by doing a "SYSDUMP" to a null device ($NULL). This particular program will then "ABORT" after obtaining the I/O listing. Please note that this section of the program will abort as part of its normal operation.

d. Run DPAN2.PUB.SYS to obtain a listing of the dump of your problem on the HP3000 SERIES II.

CAUTION-If the dump to be taken using this STREAM file was not physically taken on the host machine, the LOADMAP and I/O configuration generated will be incorrect. Please insure proper administrative measures to assure that the LOADMAP and I/O configuration are the correct ones for the dump in question.

This STREAMed job logs onto the account SYS using the MANAGER user. It is important to remember to remove passwords from this account/user or to modify the STREAM file "DUMPJOB" using the EDITOR subsystem.

The second file (program file) is called "FORMGEN." This program will generate twenty copies of the Software Maintenance Request form for your use to report subsystem problems and enhancements. The program output is automatically directed to a line printer in the device classs "LP." The output can be directed to another list device by using the formal file designator "LIST."
For example:

```
:FILE LIST;DEV=$STDLIST
```

The above example would direct output to the session/job output device which could be a hardcopy terminal.

We hope that the use of these two files will simplify your procedures for reporting problems to HEWLETT-PACKARD.

II. SUPPORTED UTILITIES

A. UTILITIES MODIFIED A.01.XX

UTILITY	CHANGE HISTORY A.01.XX													
	01	02	03	04	05	06	07	08	09	10	11	12	13	14
DISKEDT2	X													
DPAN2	X													
FREE2	X													
LISTDIR2	X													
LISTEQ2	X													
LISTLOG2	X													
PATCH2	X													
MEMLOGAN	X													
MEMTIME	X													
SADUTIL	X													
SLPATCH	X													
SPOOK	X													
RECOVER2	X													

B. ENHANCEMENTS

1. LISTEQ2 has been enhanced to print up to 96 file equations. This increases the limit from about 24.

2. LISTLOG2 now runs in batch as well as session mode.

3. All of the SUPPORTED UTILITY modules have been modified to output a COPYWRITE statement. They will also identify their fix level now.

4. DPAN2 has been enhanced to issue a rewind to the mag tape prior to attempting to read it. This will also make DPAN2 more compatible to DPAN on the CX.

C. CORRECTIVE SOFTWARE CHANGES

1. LISTLOG2 has been modified to output the LOGON queue from the LOGON record.

2. DPAN2 has been corrected to eliminate all reported occurences of INTEGER OVERFLOW as well as correcting the problem of printing the correct BANK and page number.

D. KNOWN PROBLEMS

RECOVER2 will not recover any files with privileged file code. In particular, DATA BASE files created through

IMAGE cannot be recovered. Until this is fixed, ensure that historical backup of these files is done operationally at regular intervals.

SPL HP32100A.06.04

DATE CODE 1709, N00N100A.HP32100.SUPPORT

A. ENHANCEMENTS

1. A running count of the error and warning number is output with each error or warning respectively.

2. An EDITOR QUICK file may be used as the TEXT or MASTER (or both).

3. The execution times of the SERIES II will be listed when the INNERLIST option is specified on a SERIES II.

4. If the $NULL option is chosen for the LIST file then no call to FWRITE will be made to the LIST file. Error and warning messages will be listed to $STDLIST along with the compilation summary in this case.

5. If the USL file exhausts the available directory space then a call is made to ADJUSTUSLF so that the directory portion of the USL file is enlarged. If the end of file is encountered while writing to the USL file then a call is made to EXPANDUSLF to increase the size of the file. Heretofore the compilation was terminated with the error message "USL OVERFLOW".

B. CORRECTIVE SOFTWARE CHANGES

1. The number of warnings was incorrect as output to the $STDLIST file if the LIST file was not $STDLIST. The count was truncated by one digit on the right. This has been corrected.

2. The USL file was opened with 400 sectors even if it was specified to be of larger size in a FILE command. The USL file is now opened with the size specified.

3. A procedure using the INTERRUPT option caused the USL file to be unusable. This has been corrected.

4. If the absolute value of a LONG variable is stored into another LONG variable the code for storing into the second variable was missing. This has been corrected.

5. A double quote ("") in a DEFINE declaration was erroneously compressed into a single quote which resulted in an invocation error. Double quotes are no longer compressed in DEFINE declarations and are compressed only when the DEFINE is invoked.

6. S-relative addressing was incorrect when a REAL variable was converted to a LONG variable. This has been corrected.

7. If a procedure call is made to another procedure which is option VARIABLE and there are 16 parameters the first of which is present then the mask word was incorrect. This has been corrected.

8. A missing semicolon following a statement within a CASE statement was not detected as an error. This has been corrected.

9. A compare range and branch instruction nested in a set of OR operators resulted in executing the conditional statement under the wrong conditions. This has been corrected.

10. Superfluous left parentheses within the boolean expression in a conditional statement were not flagged as errors. This has been corrected.

11. An EXTERNAL variable used as the loop variable in a FOR statement resulted in an incorrect address assignment for that variable. This has been corrected.

12. A MOVE WHILE statement using a PB-relative array as the source was undetected as an error condition. This has been corrected. (There is no PB option for this instruction.)

13. The index calculation in a MOVE statement was incorrect if the array used was of type LONG. This has been corrected.

14. An array declaration using a PB array for a reference resulted in an erroneous ADR line of output. This has been corrected.

15. An error occurring in a DEFINE declaration could cause the remaining source records to be flushed. This has been corrected.

16. The instructions "MABS", "MFDS", "MTDS", and "MDS" have an allowable maximum SDEC field of 7. The previous version of SPL did not allow the maximum to be used in an ASSEMBLE statement. This version does.

17. S relative addressing was incorrect when intrinsics with value long parameters were used. This has been corrected.

18. A secondary entry point in a procedure was not ceased during a recompilation of that procedure into a USL file when the procedure name and the entry name were hashed into the same bucket. This has been corrected.

19. Heretofore, an OWN LONG POINTER declaration was flagged as an error. This is no longer the case.

20. A negative constant used as an index to a long variable resulted in a positive index. This has been corrected.

21. A byte array which was initialized with a list of constants which exceeded 2038 bytes caused an extraneous word to be inserted into the array. This has been corrected.

22. An error during an attempt to close a file caused the compiler to loop. This has been corrected.

23. Dynamic bounds for a long array resulted in erroneous code generation for allocating the array. This has been corrected.

BASIC INTERPRETER, HP32101B.00.06

DATE CODE 1709, N00N101B.HP32101.SUPPORT

A. CORRECTED SOFTWARE CHANGES

The "prep-time" error message "INCOMPATIBLE DIMENSIONALITY IN USER FUNCTION REFERENCE" subsequently caused anomalous results when the error occurred in a program which had been named (via the NAME or GET command).

B. KNOWN PROBLEMS

BASIC aborts with a stack underflow when control-y is typed in certain circumstances. This occurs most often when printing the FREQ table. This sometimes arises when INVOKing or using the ABORT, CALLS or FILES commands in BREAK-mode. [BR #1396].

Work-around (for BREAK-mode commands): Type control-y and set a BREAK at the next statement to be executed. Then enter the GO or RESUME command. When you break at that statement, it will then be safe to use any BREAK-mode commands.

FORTRAN HP32102B.00.07

DATE CODE 1709, N00N102B.HP32102.SUPPORT

A. ENHANCEMENTS

1. Addresses in the symbol MAP will be octal addresses (preceded by a '%') instead of decimal. This was changed so as to facilitate debugging since code and stack dumps are represented in octal.

2. Error #50 has been changed from

"PROGRAM UNIT ABORTED -- OUT OF SPACE"
to
"PROGRAM UNIT ABORTED -- MORE DL-DB SPACE NOT AVAILABLE"

This message is given when the compiler tries to expand its DL to DB area of the stack, which is used for the symbol table and other necessary tables and buffers, but was not able to obtain this additional main memory due to the current system environment.

B. CORRECTIVE SOFTWARE CHANGES

1. A cross reference was not produced when there was an error in the compile. In this version a cross reference will always be made if specified.

2. FORTRAN did not recognize a function which was typed INTEGER*4.

3. The sequence

```
$CONTROL FILE = nn (or nn-nn)
 Block Data
 Main Code
```

did not cause the logical units specified by "FILE =" to be entered into the Fortran Logical Unit Table (FLUT). This problem occurred because the buffer which contains this information is re-initialized to zeros at the beginning of each program unit. Since the Block Data subprogram is not able to specify the contents of this buffer in the USL file, this information was lost. In this version of FORTRAN the Logical Unit Table buffer is not re-initialized if the previous program unit was Block Data.

4. Passing an array of type other than INTEGER or LOGICAL at a primary entry point, but not the secondary entry point, resulted in a bounds violation when the program was executed. When this parameter is an array, FORTRAN modifies this address for internal reasons. A mistake was made in handling this particular case.

5. Type conversion from COMPLEX to INTEGER*4 resulted in incorrect code.

C. DOCUMENTATION CHANGES

1. All data types other than CHARACTER are stored in multiples of whole 16-bit computer words. CHARACTER values are stored in multiples of 8-bits (two 8-bit characters (bytes) per word). Addresses will be octal word addresses except for the type CHARACTER which will have an octal byte address (since CHARACTER values do not necessarily start on word boundaries).

   Variables which are contained in a COMMON block will show addresses in the symbol MAP which are relative to the beginning of that COMMON block (since the DB relative location will not be determined until PREP time).

2. If dynamic array bounds are used in a subprogram which contains secondary entry points, the dynamic bounds must be passed in each entry point even though the code being executed does not access the array. This is because the space must be allocated for the array even if it is not used. Since the compiler does not check to make sure the bounds are passed at each entry point, failure to do so causes the program to abort with a bounds violation. The violation appears in the subprogram's initialization code when the calling program makes an entry at one of the points that does not include the necessary dynamic bounds variables.

3. A FORTRAN/3000 subroutine containing a Trap statement may not be added to the System Library since the trap handling mechanism requires a special COMMON to store the external label of the called subroutine (plabel).

4. It is important to note that FORTRAN types CHARACTER and COMPLEX are not the same as SPL types BYTE and TWO ELEMENT REAL ARRAY but the correspondence is possible since the compilers implement such types similarly. When passing parameters requiring the correspondence of CHARACTER to BYTE or COMPLEX to TWO ELEMENT REAL ARRAY from a FORTRAN procedure to an SPL procedure, do not use OPTION CHECK 3 in the called SPL procedure. This option directs the Segmenter to check the procedure type, number of

parameters, and type of each parameter for exact correspondence. Since the types COMPLEX and CHARACTER are not available in SPL, the use of OPTION CHECK 3 results in a segmenter error. To avoid this problem use a CHECK number of 2 or less.

D. KNOWN PROBLEMS

1. The displacement within a COMMON block for an INTEGER*4 or DOUBLE PRECISION array is not computed correctly in the symbol map.

2. Incorrect code is generated if an input or output statement (i.e. read, write, accept, display) is given as an argument an expression containing a substring designator for a character variable which contains another character variable or character string. Example:

```
DISPLAY CH[1:INDEX(CH,' ')]
```

does not work properly.

3. It appears that CROSSREF ALL may cause a very large compile to abort.

BASIC COMPILER, HP32103B.00.04

DATE CODE 1709, N00N103B.HP32103.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

1. When a parameter in a CALL was an integral constant, the corresponding descriptor indicated that an integer was passed when, in fact, a real number is passed.

2. When a simple string variable without a substring designator, "(*)", appeared in the call to a user-defined function or CALL'ed routine the string was sometimes passed by-value instead of by reference depending on the statements immediately preceding the call.

3. A "bounds violation" run-time abort resulted if the array in a MAT NUL$ statement within a user-defined function corresponded to a function parameter which was not in COMmon.

4. Incorrect results or a run-time abort occurred when executing a MAT Scalar Multiply statement in which the scalar expression consisted of a simple or subscripted REAL variable and the array was type-LONG.

5. Anomalous results or a run-time abort occurred when executing a call to a user-defined function in some cases. The problem arose when the call was within another user-defined function which appeared in the program source after the function being called. Incorrect code was generated only when certain boundary conditions were detected at compile-time.

6. Anomalous results sometimes occurred when the branch to the label on a string-to-numeric CONVERT statement was taken; a common symptom might be incorrect execution of an enclosing FOR-loop In particular, the branch:

   a. failed to properly exit an enclosing FOR-loop if the label was outside the FOR-loop

   b. failed to properly clean-up if the string was a string expression.

7. Anomalous results or run-time errors occurred when executing very complicated AND-OR conditional expressions in IF statements The problem arose only when rare boundary conditions were detected at compile-time.

8. TYP(0) returned 5 (integer) for integral constants rather than 1 (real).

9. A run-time abort due to an integer overflow resulted for some string array references where the subscript was a non-constant. The problem arose when the subscript multiplied by the element size (in bytes) exceeded 32767.

10. The compiler aborted with a bounds violation when compiling a MAT PRINT USING statement which contained a string array item and in which the format was specified by a string expression.

11. Due to recent changes in MPE, executing a SYSTEM statement containing a literal string command image resulted in spurious MPE command errors.

12. A spurious run-time error (#86) resulted in some cases when an ASSIGN opened a file for a reserved file position ("*" in the FILES statement). The error arose when the Global File Number of the file position had previously been assigned to a linked file ("#<integer>" in the FILES statement). This might occur only when the INVOKE statement was used.

B. KNOWN PROBLEMS

1. When the base is type-REAL and the power is a type-LONG constant representable as an integer (eg., 9**(-2L0)), single-precision rather than double-precision arithmetic is performed. [BR #2007]

   Work-around: replace type-LONG constant power with a variable

2. The unary-minus operator preceding a constant is not always handled correctly. This causes the following two incorrect results (where "x" is a constant and "y" is any variable, constant or expression):

   a. "-x MOD y" is evaluated as "(-x) MOD y" instead of "-(x MOD y)"

   b. "-x**y" is evaluated as "(-x)**y" instead of "-(x**y)". [BR #2008]

   Work-around: Fully parenthesize expression.

3. Incorrect code is generated for I/O FOR-loops with a constant negative one STEP size. [BR #6006]

   Work-around: Replace the constant STEP size with a variable.

4. Incorrect code is generated for a FOR-loop enclosing both an ONEND statement with a destination within the loop and a GOTO statement with a destination outside the loop. This will cause spurious run-time aborts if an end-of-file is detected while inside the FOR-loop.

   Work-around: Place a superfluous GOTO outside the FOR-loop with a destination inside the loop. The GOTO statement itself is not intended to be executed.

5. Lower-case characters are not recognized as format specifiers in PRINT USING format string expressions.

6. A bounds violation or other anomalous results occur when a user-defined function is used in a subscript expression on a left-hand side of a LET statement. For example:

```
X(FNA(Y))=10
```

Work-around: Eliminate the reference to the user-defined function in the subscript expression by evaluating it in a preceding statement. For example:

```
Z=FNA(Y)
X(Z)=10
```

C. MISCELLANEOUS

Version HP32103B.00.03 was skipped due to a problem which was introduced in that version and was fixed subsequently just prior to this release.

RPG SUBSYSTEM HP32104A.03.04

DATE CODE 1709, N00N104A.HP32104.SUPPORT

A. ENHANCEMENTS

1. Programs using System/3 line skipping may now line up channel one with line one. RPG will assume it is starting at line 1 instead of line 6 if a 1 is put in column 53 of the header record. (This is the column an L had been put in previously).

   A 1 in column 53 of the header record indicates System/3 type line counting starting at line 1.

2. The RPG compiler loads and stores the line number for each statement it executes, so that when a run time error appears RPG can inform the user where the error occurred. If this feature is not needed in a program the user can save up to 30% of code space (resulting in faster execution, especially of large programs) by entering an N in column 20 of the header record.

   An N in column 20 of the header record indicates that statement numbers will bot be available during program execution.

B. CORRECTIVE SOFTWARE CHANGES

1. An extra byte was blanked out if edit code required a zero field to be printed as blanks.

2. Output files were opened as outkeep causing regular files not to be rewound and RSAM files a problem with file sharing.

3. Very large programs (over 1000 symbols) cuased bounds violation when loading compile time arrays.

4. Variable array indices defined after they were used caused bounds violation when loading compile time arrays.

5. When alternating tables/arrays were not voth tables or arrays, no message was given. This is now a fatal error (537).

6. With very large programs the cross reference could have aborted with a bounds violation.

7. Skip after to current line at total time when current line was overflow line resulted in RPG adding two blank lines and not counting them.

8. Computation of an array index in internal subroutine before use sometimes caused wrong index to be returned or bounds violation.

9. Processing an IMAGE DS sequentially between limits and the limit specified in the RAF were OUTSIDE THE ENDS of the dummy chain caused all DS records in the chain to be mistakenly accepted (for both forward and backward sequential reads).

C. DOCUMENTATION CHANGES

The second edition of the RPG/3000 manual is now available. This new edition includes many of the new features of RPG added since the last edition. These features include KSAM/3000 and RSAM/3000, partial field translation, locking and unlocking shared files to control access, file-sharing, and features to allow increased compatibility with System/3 RPG II. It is recommended that all users of RPG/3000 obtain a copy of this new edition for their reference.

APL HP32105A.0.03

DATE CODE 1709, N00N105A.HP32105.SUPPORT

A. ENHANCEMENTS

1. Changes have been made to the file system access to make it faster.

2. Scan has been greatly speeded up for plus, minus, times divide, MAX, MIN, AND and OR.

B. CORRECTIVE SOFTWARE CHANGES

1. The ability to suspend APL by hitting the BREAK key has been disabled. Use the)MPE command to switch from APL to MPE. This was done since on most APL systems the break key is used to suspend an APL function. APL\3000 uses control Y.

2. Setting)DEPTH to a value greater than 1092 would abort APL in version .02. This is fixed.

3. There was a bug in the interrupt handler which put APL into a special debug mode indicated by a '>' being typed out if eleven control Ys were hit. This was fixed and the system was modified so that, after twenty control Y signals in a row, APL will save the 'A' file and restart. Hopefully, it will be 'impossible' to get in twenty control Ys before APL responds with an interrupt. Please file a bug report if this happens. It will be necessary to)COPY the 'A' file to recover the workspace.

4. During an initialization of APL one of the micro-instructions is executed. This will cause APL to imediately abort rather than generating an extraneous 'A' file before aborting in the case that the APL microcode is not installed.

5. A problem with workspace initialization was fixed. This problem appeared when a workspace overflowed and was copied into a larger file.

6. Some data variables were being shared when they should have been covered by parameter passing in a function call. This is fixed.

7. The definiton of []WA has been changed. It gives the number of bytes used by the current workspace,not the amount of workspace left. It was felt that the standard definition of []WA had little meaning with a virtual workspace, whereas telling the user the size of the workspace file did convey some useful information.

8. Functions which had a large proportion of comments could put APL in an infinite loop upon ENDing an EDIT or using []FX. This was fixed.

9. If an argument to []CR was longer than 128 characters a 'bounds violation' would result. This no longer happens.

10. If a ")" command was given as a response to QUAD input the system would try to execute it without much success. Now, inputting one of these ")" commands will cause an error message with no effect on APL.

11. The following problems have been corrected, and are listed in operator order.

a. Scalar dyadic

1) 0?0 created a problem which later could cause APL to abort. This has been fixed.)COPY workspaces where 0?0 may have been executed.

2) Arithmetic Progression Vector times 0 caused an infinite loop.(i.e. 0 TIMES iota 10)

3) Scalar dyadic operator on beaten APV expression caused an infinite loop.Example: 0 MAX (complex beaten APV expression)

4) Scalar dyadic operator on logical/integer expression combination occasionally gave bad answers (all zeros or ones) Example:(logical expression) minus (integer expression)

b. REDUCTION and SCAN

1) Multiple reductions in an expression sometimes produced bad results on re-execution. Example:+/+/(expression)...second time.

2) Logical operator reduction over complex beaten subscript expression gave wrong answers. Example:OR/(subscript expression)AND(subscript expression)

3) Reduction over logical expression involving reduction gave wrong answers. Example: +/0 1 TIMES +/(expresssion)

c. DECODE

Beaten expressions involving decode gave wrong answers unless a temporary assignment was used.

d. OUTER PRODUCT

+/[1]outer product aborted APL when the matrix was too large. Two fixes were made. One which increase the size of the arguments allowed in the outer product. The other which catches the size error and returns a message rather than aborting APL.

e. REVERSAL

The reversal of some beaten large expressions aborted APL. Example:REVERSE RHO(very large rank 3 expression)

f. TAKE

1) First time: >1 TAKE unit; second time: 0 TAKE of unit caused infinite loop on second execution.

Example: Function with N TAKE 7..First N is 3, then N changes to 0.

2) TAKE where the number of resulting elements was >=32767 aborted APL. This no longer happens.

g. CONCATENATE and LAMINATE

CATENATE and LAMINATE of Arithmetic Progression Vectors aborted APL. Example: APV,[.5]APV

LAMINATE where either argument was scalar or a single element array gave incorrect results or aborted APL.

CATENATE where the total number of resulting elements was >32767 aborted APL. An error message now results.

h. EXPAND

The expansion of a complex beaten expression involing reversal sometimes gave incorrect answers. Example: 1 0 1\2 TAKE REVERSAL (expression)

i. APLGOL

Very long constant vectors or comments would cause a bounds violation. This was fixed.

Using calculator mode,QUAD input or EXECUTE in)LANG APLGOL caused problems which eventually lead to a system error. Workspaces with APLGOL functions or in which APLGOL may have been used in calculator mode should be)COPY'ed.

C. KNOWN PROBLEMS

1. Function LOCKing is not yet implemented.

2. File LOCKing does not work.

D. MISCELLANEOUS

In addition to the program file, APL requires a set of PROMS mounted on the EIS board. This contains the extra instructions which APL executes. Without these instructions an unimplemented instruction error will result.

The following two items are helpful definitions:

1. Arithmetic Progression Vector (APV)--The data structure used to represent simple integer vectors. It consists of three integers--start, increment and length.

2. Beaten Expression--An expression for which the code was optimized by the compiler. TAKE, DROP, REVERSAL, TRANSPOSE, SUBSCRIPT are some functions which are beaten.

The following are suspected bugs:

3. Some possible problems have been observed in or related to, the following two items, but not enough evidence has been gathered for verification. We would appreciate your observations on:

   a. Setting []VM to an illegally large number of pages.

   b. Loss of control Y.

DISTRIBUTED SYSTEMS 3000 HP32190.00.00

DATE CODE 1709, N00N190A.HP32190.SUPPORT

A. KNOWN PROBLEMS

1. Remote remote does not work.

2. An occasional ERR 0 occurs that should not happen.

3. ERR 246 when user on system A does a "DSLINE;CLOSE" between user on system B's DSLINE and remote HELLO.

4. About 10% of the time a "BREAK" in PTOP while the slave is doing STDIN/STDLIST will cause RESUME to have no effect.

5. "DSLINE @;CLOSE" with many opens is peculiar, and a "NO" response to "ABORT REMOTE?" does not work.

6. "=DSLINE LDEV< SHUT" after :DSLINE on SSLC hangs the console until the phone is dialed.

7. If session is on the console and a =DSLINE is issued, can't get a control A until the session does something.

8. If in the "#REMOTE" subsystem the # prompt is issued in response to a "remote HELLO", a "CR" or some similar activity is needed before the error message "ERR 60,216" is issued to indicate no virtual terminals are available.

9. "BREAK" during a "BYE" message hangs the terminal.

10. Occasional system failures may result because of:

    a. FWRITE of a TCOUNT which contains a "garbage value".

    b. Attempted FWRITEs beyond EOF.

    c. Continuation records specified and number of records would overflow an extra data segment.

B. MISCELLANEOUS

Following are suspect problems to be investigated:

1. Remote RJE.

2. Break in APL

3. Control Y in APL.

4. Master and slave sessions sometimes cannot be aborted.

EDITOR HP32201A.06.01

DATE CODE 1709, N00N201A.HP32201.SUPPORT

A. ENHANCEMENTS

A tab feature has been added to the ADD and REPLACE commands. The tab feature provides more rapid entry of text by allowing certain columnar positions to be identified as "tab stops". A tab character is "expanded" only when lines of text are put into the WORK file by an ADD or REPLACE command. "Expansion" is done by replacing a tab character with one or more blanks, depending on the positions of the tab character and the "tab stops".

Following is a description of the tab feature:

1. The tab feature is initially disabled.

2. To specify tab options, the SET command is used.

   S[ET] TABS [= (colnum [, colnum]...)]

   S[ET] TABCHAR [= string]

As many as 12 tab stops may be set. Each "colnum" must be larger than the preceeding "colnum", and must lie between LEFT and RIGHT values. The length of "string" may not be greater than 1 character. A decimal value for "string" may be given by preceeding the digits with an apostrophe.

a. /SET TABS

   Causes the default tab stops (4,7,10,13,16,19,22,25,28,31,65,71) to be set. When FORMAT is COBOL the default tab stops are (6,10,14,16,22,26,30,34,38,46,54,67).

b. /SET TABS=(11,22,33,44,55,66)

   Causes tab stops to be set at the columns indicated.

c. If tab stops are enabled, the default tab character is CONTROL I, which is the ASCII HT, or decimal 9.

d. /SET TABCHAR="!"

   Causes ! to be the effective tab character, to override the default CONTROL I. This could be used to define a printing character for terminals that do not support the ASCII tab character.

e. /SET NOTABS

   Causes the tab feature to be disabled.

3. To display tab options, the VERIFY command is used.

   a. /VERIFY TABS

      Shows the column numbers where the tab stops are set.

   b. /VERIFY TABCHAR

      Shows the tab character currently in effect. If the tab character is non-printing the decimal value is printed, preceeded by an apostrophe.

Following are examples of the tab feature use:

```
:EDITOR

HP32201A.06.01 EDIT/3000  THU, MAR 31, 1977,  8:02 AM
(C) HEWLETT PACKARD CO. 1976
```

```
/V TABS
NO TABS USED
/
/S TABS
/
/V TABS,TABCHAR
TABS = (  4,  7, 10, 13, 16, 19, 22, 25, 28, 31, 65, 71)
TAB CHARACTER = '9
/
/S TABS = (11,22,33,44,55,66), TABCHAR = "$"
/
/V TABS, TABCHAR
TABS = ( 11, 22, 33, 44, 55, 66)
TAB CHARACTER = "$"
/
/
/A
    1      $1$2$3$4
    2      $A$B$C
//
/L ALL
    1                1          2          3          4
    2                A          B          C
/
/S TABCHAR='62
/V TABCHAR
TAB CHARACTER= ">"
/
/E
```

Following is a description of using TABS with the 2640 series terminals:

The HP 2640 series and some other terminals have "TAB SET", "TAB CLEAR" and "TAB" keys. These can be used together with the tab feature to allow the line being entered to show the effect of tab stops while it is being typed in. To do this, press the "TAB SET" key after positioning the cursor at each of the columns where tab stops are desired. These positions will depend on the EDIT/3000 command being used, since the ADD and REPLACE commands print 10 characters in front of each line while the ADDQ and REPLACEQ commands do not.

There are some issues in the use of the TAB key that the user may need to be aware of to avoid potential confusion. When a CONTROL H (backspace) is entered following the tab character, the cursor may not indicate the correct column position,

since the terminal only backspaces one character but the tab character may have caused several spaces to appear on the line being entered. Using the CONTROL H (backspace) key immediately after using the TAB key will delete the tab character from the input line, but will not necessarily reposition the cursor to the column where it was located before the TAB key was hit. Another issue the user needs to know is that when the TAB feature is disabled (initially or via /SET NOTABS) the Editor will not "expand" the tab character, that is, the tab character will not be converted to blanks but will be put in the WORK file as a tab character. If this should inadvertently occur, the user may do a HOLD on the line(s) with tab characters, DELETE the line(s), do a /SET TABS, and then do an ADD...,HOLD to cause the tab characters to be "expanded", or converted to blanks.

For frequently used tab stop formats, a file containing SET and Q commands can be used to set physical tab stops on the terminal. Referencing the file with the USE command will cause the Q command to print a line on the terminal. The code for "TAB SET" is "ESC 1". In any column where an ESC 1 would appear, a tab stop will be set. Be sure to reset existing tab stops with "ESC 2" or with "RESET TERMINAL".

An example of a USE file to set tab stops on a 2640 terminal, in FORMAT=DEFALT, for ADD or REPLACE (not AQ or RQ), follows:

```
/S TABS=(4,7,10);
/Q" E";                    << the E is preceeded by ESC >>
/Q"          1  1  1";     << each 1 is preceeded by ESC >>
```

The ESC E will reset any existing terminal tab stops. Each ESC 1 will set a terminal tab stop in the column where it is placed.

B. CORRECTIVE SOFTWARE CHANGES

1. When Z:: is active, an "&" is not interpreted as an END command.

2. XPLAIN ALL from a job will not cause a file error.

3. Entering a record longer than the work file record length will not cause a file error. A warning message is issued and the line is truncated to the work file record length.

4. The LENGTH value cannot be greater than the current work file record length. The value of length can be as large as 255 if the work file has not yet been built.

5. The DELETE command correctly deletes the first record in a range when LEFT is greater than 1.

6. ADDQ correctly suppresses line number output in a job or in a session when doing an ADDQ from HOLD when BATCH is set.

7. The message "CLEAR HOLD?" is consistently issued when a HOLD is requested and the APPEND option is not used.

8. A condition that caused an abort due to a bounds violation when a number greater than 255 was used as a column number has been corrected.

C. KNOWN PROBLEMS

1. Any error within a USE file will cause the Editor internal flag to be set FALSE. Subsequent commands will be skipped until a YES command, or Y command, is entered.

2. Exhausting available work file space during a GATHER, INSERT, or JOIN command will cause a file error.

3. The command JOIN (#recnum/#recnum) counts records beginning from one, but TEXT (#recnum/#recnum) counts from zero.

4. When LEFT is set to a value greater than 1, a VERIFY ALL can cause a file error and an abort, under special circumstances.

5. Use of the CONTROL Y during a MODIFY is ambiguous: it may be interpreted to restart the MODIFY using the original line value; also, it may be interpreted to exit from the command.

6. Re-setting the SIZE value may inhibit texting a file in its entirety, if SIZE is smaller than the TEXT file length.

7. Under special circumstances, a request to INSERT between lines previously created by INSERT can cause a file error.

8. If an INSERT exhausts available line numbers between two preexisting lines, the character string being carried forward by INSERT is lost.

9. The value of LAST used as a column position in a CHANGE command is the column position of the last non-blank character in the current line before the CHANGE command was entered rather than the line to which the CHANGE command is directed.

DEL/3000 HP32206A.01.03 (FORMAINT)

DEL/3000 HP32206A.01.04 (LIBRARY PROCEDURES)

DATE CODE 1709, N00N206A.HP32206.SUPPORT

A. ENHANCEMENTS

1. FORMAINT has been changed for use on the HP3000 Series I.

2. The procedures READTERM and WRITETERM have been modified for performance improvement in this release.

3. FORMAINT has been altered to recognize the f8 key during form input processing. The f8 key is now treated as a return to the start screen throughout FORMAINT.

B. CORRECTIVE SOFTWARE CHANGES

1. A user program with a large stack (greater than 16K words) will no longer abort with a bounds violation when calling OPENTERM or OPENFORM.

2. FORMAINT will no longer accept blank forms on input and the integer overflow when working on a form file containing a blank form definition has been fixed.

3. FORMAINT has been changed so that when attempting to access the same form file from more than one session the formfile name will be blanked out in those sessions unable to gain access to the form file. This will prevent any possible illegal access to the form file.

C. KNOWN PROBLEMS

1. When using FORMAINT with a Series I, during form input for form creation or modification any line longer than 216 bytes will cause a read timeout on a Series I system. This timeout is recognized by FORMAINT and an appropriate error message is generated. This change does not affect Series II operation.

2. DEL will not run properly on CX or SERIES I when stand alone at 2400 baud on the 2640B. This is because on the SERIES I, the terminal driver does not use ENQ/ACK when sending over 80 characters before more characters are sent. GET AROUND: change baud to 1200 on 2640B.

KSAM HP32208A.00.04

DATE CODE 1709, N00N208A.HP32208.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

1. When keyblocks are higher than 2 levels and use FFINDBYKEY with relop=1 or 2, the current record pointer might not point to the correct place in the file.

2. Variable lenght KSAM file does not allow user to enter more than N records, where N is number in the DISC=N,P,Q keyword in the BUILD command.

B. ENHANCEMENTS

LOCKING/UNLOCKING KSAM FILES FROM COBOL

Beginning with version A.00.04 of the KSAM/3000, the KSAM files may be dynamically locked and unlocked from COBOL programs. Before locking a KSAM file, it must be opened with dynamic locking and share access enabled by call the new open procedure CKOPENSHR.

The actual locking and unlocking must be done, after calling CKOPENSHR, by calling the new procedures CKLOCK and CKUNLOCK.

1. CKOPENSHR

   The calling sequence and paramenters are the same as CKOPEN except the KSAM file is opened with dynamically locked and share access enabled.

2. CKLOCK

   This procedure may be used to lock a KSAM file which has already been opened with a call to CKOPENSHR.

```
!-------------------------------------------------------!
!                                                       !
!      CALL "CKLOCK" USING filetable,status,lockcond    !
!                                                       !
!-------------------------------------------------------!
```

PARAMETERS

filetable: an 8-word record containing: the name of the ksam file, its input-output type, access mode, the filenumber given tne file when it was last opened, and a code indicating whether the previous operation on the file was successfull and if so what it was. (Refer to filetable parameter discussion in the KSAM/3000 manual)

status: one-word (two 8-bit character) set to pair of values upon completion of the call to CKLOCK. It indicates whether or not the file was successful locked and if not, why not. The two characters are set to "00" if CKLOCK is successful, to "30" if request denied because the file was locked by another process, to "9n" if not successful and n is the file system error code in binary.

lockcond: this parameter corresponds to the second paramenter of the FLOCK intrinsic; that is, an odd (TRUE) value specifies that if the file cannot be locked immediately, the process suspends until it can be locked. An even (FALSE) value will return to the calling process if the file cannot be locked.

EXAMPLES

```
77 LOCKCOND              PICTURE S9(4)  COMP   VALUE 1.
77 RESULT                PICTURE  9(4)         VALUE 0.
01 STATUSKEY.
   02  STATUS-KEY1       PICTURE  X            VALUE " ".
   02  STATUS-KEY2       PICTURE  X            VALUE " ".
```

```
01 FILETABLE.
   02  FILENUMBER        PICTURE S9(4)  COMP  VALUE 0.
   02  FILENAME          PICTURE  X(8)        VALUE "KSAMFILE"
   02  I-O-TYPE          PICTURE S9(4)  COMP  VALUE 0.
   02  A-MODE            PICTURE S9(4)  COMP  VALUE 0.
   02  PREV-OP           PICTURE S9(4)  COMP  VALUE 0.


    CALL "CKLOCK" USING FILETABLE, STATUSKEY, LOCKCOND.
    IF STATUSKEY = "00"
       THEN DISPLAY "CKLOCK IS A OK"
    ELSE IF STATUSKEY = "30"
            THEN DISPLAY "FILE LOCKED BY ANOTHER PROCESS"
         ELSE IF STATUS-KEY1 = "9"
                 THEN CALL "CKERROR" USING STATUSKEY, RESULT
                      DISPLAY "ERROR NO.", RESULT.
```

3. CKUNLOCK

This procedure may be used to unlock a KSAM file.

```
!-------------------------------------------------------!
!                                                       !
!     CALL "CKUNLOCK" USING filetable,status            !
!                                                       !
!-------------------------------------------------------!
```

PARAMETERS

filetable: an 8-word record containing the name of the ksam file, its input-output type, access mode, the filenumber given the file when it was last opened, and a code indicating whether the previous operation on the file was successfull and if so what it was. (Refer to filetable parameter discussion in the KSAM/3000 manual)

status: one-word (two 8-bit character) set to pair of values upon completion of the call to CKUNLOCK. It indicates whetner or not the file was successful UNlocked and if not, why not. The two characters are set to "00" if CKUNLOCK is successful, to "31" if request denied because the file was not locked by this process, to "9n" if not successful and n is the file system error code in binary.

EXAMPLES

```
77 RESULT                PICTURE  9(4)        VALUE 0.
01 STATUSKEY.
   02  STATUS-KEY1        PICTURE  X           VALUE " ".
   02  STATUS-KEY2        PICTURE  X           VALUE " ".
```

```
01 FILETABLE.
   02  FILENUMBER        PICTURE S9(4)  COMP  VALUE 0.
   02  FILENAME          PICTURE  X(8)        VALUE "KSAMFILE"
   02  I-O-TYPE          PICTURE S9(4)  COMP  VALUE 0.
   02  A-MODE            PICTURE S9(4)  COMP  VALUE 0.
   02  PREV-OP           PICTURE S9(4)  COMP  VALUE 0.


CALL "CKUNLOCK" USING FILETABLE, STATUSKEY.
IF STATUSKEY = "00"
   THEN DISPLAY "CKUNLOCK IS A OK"
ELSE IF STATUSKEY = "31"
        THEN DISPLAY "FILE NO PREVIOUS LOCKED BY THIS PROCESS"
     ELSE IF STATUS-KEY1 = "9"
             THEN CALL "CKERROR" USING STATUSKEY, RESULT
                  DISPLAY "ERROR NO.", RESULT.
```

FCOPY, HP32212A.1.04

DATE CODE 1709, N00N212A.HP32212.SUPPORT

A. ENHANCEMENTS

The previously announced enhancement for HP2644A terminals applies only to the HP3000 Series II because of dependencies upon enhancements to MPE in the file system.

B. CORRECTIVE SOFTWARE CHANGES

An abort with a bounds violation segment 0 location 3253.

COBOL HP32213C.01.04

DATE CODE 1709, N00N213C.HP32213C.SUPPORT

A. ENHANCEMENTS

1. COPYLIB files can be KSAM files.

2. The size of the virtual symbol table was enlarged.

3. The following improvements have been added to the Cobol Library in version 4.06:

   a. The error message for Error 630 has been changed to ERROR DURING WRITE OF USER LABEL. A File Information Displa will also be generated with an Error 630.

b. The error message for Error 631 has been changed to USER LABEL SPACE UNALLOCATED OR ATTEMPT TO WRITE BEYOND LABE LIMIT.

B. CORRECTIVE SOFTWARE CHANGES

1. Lower case letters in PICTURE clauses were not upshifted.

2. An ON SIZE ERROR clause where the first operand had fewer digits than the result sometimes produced an Error 210 or 211 at compile time.

3. The ACCEPT statement did not work for a table element.

4. A COMPUTE statement at the start of the Procedure Division did not always work properly.

5. The compiler could not recover from an 88-level in a MOVE statement.

6. A GO TO DEPENDING ON statement nested within a conditional clause (such as an AT END clause) produced Error 211 messages--- MULTIPLY OR UNDEFINED INTERAL LABEL.

7. An ADD...TO statement with multiple results of USAGE COMP-3 or DISPLAY left two words on the stack and could produce a stack overflow abort or control flow problems with PERFORM or GOBACK statements.

8. If more than two index names were specified in an INDEXED BY clause, only the first two were handled properly.

9. ADD CORRESPONDING of DISPLAY items aborted at run time.

10. A 4-digit COMP-3 item used as a subscript or compared with an index-name did not work properly.

11. A VALUE clause at the group level in the LINKAGE SECTION was not handled properly. A warning message was not issued and bad object code was generated.

12. Search keys sometimes produced erroneous error messages and compile time aborts in large programs.

13. Subprograms sometimes aborted with a bounds violation in the initialization procedure.

14. A numeric literal move sometimes caused the compiler to abort in large programs.

15. An ADD statement with a numeric edited or alphanumeric item somewhere other than after the word GIVING was not detected as an error.

16. Arithmetic expressions with literals occassionally generated bad code in large programs.

17. A READ statement without a file name caused the compiler to abort.

18. Several problems which produced End of File errors on COBSTAB and COBDTAB at compile time have been corrected.

19. A GO TO DEPENDING ON statement occasionally aborted at run time.

20. The symbol map in large programs with sections sometimes failed to print all section and paragraph names.

21. An attempt to perform an empty section generated Error 201. The error message has been improved to show which section was being performed and where the perform statement is located.

22. Continuation of a word did not work if the line being continued contained a completed non-numeric literal.

23. Text on a line following the period of a COPY statement caused the compiler to abort with a bounds violation or a write error on the list file.

24. Subprograms with large linkage sections sometimes aborted during the initialization procedure.

25. Using an index-name with the wrong table was not detected as an error.

26. A non-unique reference generated an Error 21 for each possible qualifier. Now only one error will be generated.

27. A GOBACK statement in a subprogram would not work if extra words had been left on the stack by some statement in the subprogram.

28. An add of 2 COMPUTATIONAL items each with less than 5 digits and both with the same number of digits after the decimal point would yield incorrect result.

29. The following problems have been corrected in the COBOL Library in version 4.06:

    a. An end-of-file condition was not detected on an ACCEPT statement.

    b. After a USE procedure was executed for a read error, the AT END branch was taken.

C. KNOWN PROBLEMS

A SEARCH statement with multiple WHEN conditions sometimes leaves extra words on the stack which can cause either a stack overflow abort or control flow problems with a PERFORM statement. This will no longer cause problems with a GOBACK statement in a subprogram (See No. 27 above).

SORT HP32214B.01.04

DATE CODE 1709, N00N214B.HP32214.SUPPORT

CORRECTIVE SOFTWARE CHANGES

Specifying an existing job temporary file as the output file in a stand-alone sort/merge aborted, instead of asking if the user wanted to purge the existing file.

IMAGE/3000 HP32215A.04.03

DATE CODE 1709, N00N215A.HP32215.SUPPORT

CORRECTIVE SOFTWARE CHANGES

DBUNLOAD has consistently reported "UNRECOVERABLE TAPE ERRORS" when in fact such errors were recoverable. The problem involved a verification procedure executed by DBUNLOAD whenever MPE reported a recovered tape write error. This problem has been corrected.

QUERY/3000 HP32216A.03.04

DATE CODE 1709, N00N216A.HP32216.SUPPORT

A. ENHANCEMENTS

It is now possible to inhibit QUERY from automatically locking the database whenever a FIND, LIST or REPORT command is executed in modes 1 or 5. To inhibit such locking, one issues the command:

```
ASSIGN LOCKOPTION = OFF
```

in response to the standard QUERY prompt ">" for command input. To resume automatic locking, the command:

```
ASSIGN LOCKOPTION = ON
```

may be issued. The current state of the locking option may be interrogated through use of the command:

```
SHOW LOCKOPTION.
```

The locking option defaults to "ON" and is reset each time a new (or the same) database is opened. This feature may be particularly useful for on-line applications hwere it is impractical to lock the database except when absolutely necessary. However, users should realize that a reduction in the level of database locking can increase the occurrence of data retrieval anomalies caused by concurrent process update.

Automatic locking is still performed during the execution of ADD, DELETE and REPLACE commands (mode 1) and this action is independent of the state of the locking option.

B. KNOWN PROBLEMS

QUERY does not accept a print position greater than 132 in a REPORT statement, even though the output is directed to a lineprinter configured to 136 characters. A temporary solution is to issue the file command:

```
FILE QSLIST;REC=-136,1,F,ASCII
```

TRACE HP32222A.03.1

DATE CODE 1709, N00N222A.HP32222.SUPPORT

CORRECTIVE SOFTWARE CHANGES

The set option did not work on the Series II because the condition code returned from a read was different under certain circumstances from the Series I. Change to TRACE will correct this situation.

CROSS ASSEMBLER (XA2100) HP32223A.01.2

DATE CODE 1709, N00N223A.HP32223.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

The Cross-Reference listing was not sorted correctly when the Assembly object code was of Absolute type. This resulted in incorrect values for ABS constants.

B. KNOWN PROBLEMS

The title which is specified in the HED statement is sometimes printed incorrectly.

2780/3780 RJE EMULATOR
HP30130C.00.03 (CS A.01.00)
DATE CODE 1709, N00N130C.HP30130.SUPPORT

A. ENHANCEMENTS

Version C.00.03 (MIT 1709) adds two new parameters to the #RJLINE command:

1. LOCK = { YES } (default=YES)
  NO

   The Emulator normally issues a LOCKSEG request for its stack; this prevents it from being swapped out. If the Emulator is run concurrently with jobs which have large memory requirements, especially in small core configurations, the other jobs may suffer severe performance degradation. This can be overridden by specifying LOCK = NO. The stack can then be handled by memory management.

2. PRI = { HIGH } (default= HIGH)
  NORMAL

   The Emulator normally issues a GETPRIORITY request for an absolute execution priority of 150 in a linear subqueue. If the Emulator is run on a high-speed (e.g., 9600 baud) full duplex line with 0 turnaround delay, the result may be that the Emulator will adversely affect the execution of other jobs, since it may be using an inordinate proportion of the CPU. This can be overridden by specifying PRI = NORMAL, which causes the call to GETPRIORITY to be bypassed. The priority is then established by the normal session or job priority.

B. CORRECTIVE SOFTWARE CHANGES

Version C.00.03 (MIT 1709) corrects the following five problems:

1. The Emulator no longer aborts a batch job when a WAIT= timeout occurs. This is true regardless of whether or not the command file is the same as the JOB input file.

2. The Emulator now restores terminals to their previous echo mode upon completion of output operations.. HOWEVER: If this version of the Emulator is run on an MPE version earlier than A01.01 (MIT 1709), the terminal will INVERT the echo mode on each output operation; this is caused by a bug in the asynchronous terminal driver, which causes the inverse of the previous echo state to be returned when an FCONTROL is issued to enable or disable echo.

3. The Emulator now issues a SYNTAX ERR 4 if an illegal value is entered for the WAIT= parameter(s). The maximum value is a total number of seconds <= 32767. If WAIT = 0 is specified, no timeout will occur.

4. The Emulator no longer tries to transmit 512 data bytes in 3780 mode; the limit is 511. This produced a buffer which was unacceptable to some host computers. EXCEPTION: If #RJIN.....;MAXSIZE = 128 is specified, the Emulator allows 512 data bytes.

5. If DIAL is specified on the #RJLINE command, and the first line operation is for output (#RJOUT, #RJLIST, or #RJPUNCH), a dial request is now issued to the operator console.

Version C.00.02 (MIT 1646/1701) corrects the following problems:

6. The Emulator previously recognized only the first component selection (routing) sequence received. This has been corrected.

7. The Emulator now allows up to 512 bytes to be transmitted in 3780 mode. It was allowing only 256.

8. If the Emulator is run from the MPE System Console, echo is no longer suppressed during output operations.

9. When using the RIN feature on #RJLINE, the Emulator previously unlocked the RIN before closing the line; this could cause another user waiting on the RIN to fail on a line open. This has been corrected.

Version C.00.01 (MIT 1640) corrects the following problem:

10. If "CONNECT = ANSWER" is not specified on #RJLINE, and a connection is not established within 15 minutes after the first transmit or receive operation, RJE will now terminate with a ****CS ERROR: 1,151 or 2,151. Previously, RJE would wait indefinitely for the connection to be established.

C. DOCUMENTATION CHANGES

The RJE Emulator manual 30000-90047 is being reprinted. The new PRI= and LOCK= parameters are included. The configuration section is reworked with more explanations. A new example using multiple forms has been added.

D. KNOWN PROBLEMS

Routed output to the Punch file is misrouted when the final block terminates with an ETX. The Punch output is routed to the Output file. The blocks which end in ETB are handled correctly.

DIAGNOSTICS HP32230A

DATE CODE 1709, N00N230A.HP32230.SUPPORT

I. MAGNETIC TAPES ASSOCIATED WITH HP32230A

Source	32230-1X001
Maintenance	32230-1X002
CPU Coldload	30000-1X016
NON-CPU C/L	30000-1X017

II. MANUALS ASSOCIATED WITH HP32230A

32230-60001
32230-60002

III. CPU DIAGNOSTICS 30000-1X016, DATE CODE 1623

SECTION	NAME	LEVEL
1	PD420A	01.00
2	PD420A1	01.00
3	PD420A2	01.00

4	PD420A3	01.00
5	PD420A4	01.00
6	PD420A5	01.00
7	PD420A6	01.00
8	PD420A7	01.00
9	PD420A8	01.00
10	PD420A9	01.00
11	PD420A10	01.00
12	PD420A11	01.00
13	PD420A12	01.00
14	PD420A13	01.00

IV. STAND-ALONE DIAGNOSTICS 30000-1X017, DATE CODE 1709

A. DIAGNOSTICS CHANGED

DIAGNOSTIC NAME	NAME	LEVEL	OCTAL FILE #
*SLEUTH	PD411A	01.01	(01)
*SDUPII	PD417A	01.01	
*CART DISC-7905A	PD419A	01.04	(02)
MEMORY PATTERN	PD421A	01.00	(03)
MULTIPLEXOR CHAN	PD422A	01.00	(04)
DISC FILE-2888A	PD423A	01.00	(05)
CART DISC-7900A	PD424A	01.00	(06)
SYSTEM CLOCK	PD425A	01.00	(07)
TERMINAL DATA	PD427A	01.00	(10)
FIXED HEAD DISC	PD428A	01.00	(11)
SELECTOR CHAN	PD429A	01.00	(12)
FAULT CORRECTING MEM.	PD430A	01.01	(13)
EXTENDED INSTRUC SET	PD431A	01.00	(14)
*NEW*HSI DIAG.	PD432A	01.00	(15)
MAGNETIC TAPE	PD433A	01.00	(16)
SSLC INTERFACE	PD434A	01.01	(17)
UI DIAG	PD435A	01.00	(20)
TERMINAL CONTROL	PD438A	01.00	(21)
CALCOMP PLOTTER	PD439A	01.01	(22)

* UPDATED/CHANGED in this MIT

B. CORRECTIVE SOFTWARE CHANGES

1. STAND ALONE SLEUTH DIAGNOSTIC

   a. A new type has been added to SLEUTH for the 7920 disc, which is 12. All commands valid for the 7905 are now valid for the 7920.

b. The CONF command has been modified to print out error messages. The command will halt only if the switch register is set up for halt on error. This should be of value when the maintenance panel is not connected to the system.

c. The following commands will now work properly with the TDI (30032 board); TDIL,RD,WD,RP.

d. An addition to CHB has been added to word shift the whole buffer (ie. the first element becomes the last element, the second element becomes the first element, etc.). This is done by specifying a "W" as the last thing in the command. This contrasts with the type "S" which is a circular bit shift of each element of the buffer (ex: CHB AA,W).

e. SLEUTH07 (7920 verifier) has been changed to check for read only.

2. SDUPII

a. SDUPII has been modified to run in batch mode so that a job stream may be used to create the diagnostic stand alone tapes.

b. The RL has two added procedures ASCII and BINARY. These procedures work exactly like the MPE's intrinsics ASCII and BINARY. For additional details see the MPE INTRINSICS manual. Procedure HELP has also been modified to do I/O to the TDI instead of the SYS CLK board.

3. STAND ALONE DISC DIAGNOSTIC

The execution of MASTER CLEAR was permanently disabled in hardware on the control board of the 7905 controller by DISC MEMORY DIVISION in 1976. All MASTER CLEARS were deleted from this diagnostic or replaced by CLEAR commands depending on the test section.

V. ONLINE DIAGNOSTICS

DIAGNOSTIC NAME	NAME	LEVEL
CARD READER	PD465A	01.00
LINE PRINTER	PD466A	01.00
2640 TERMINAL	PD469A	01.00
TERM-2762A/B	PD475A	01.00
DISPLAY TERMINAL 2644	PD477A	01.00
TERM-2615A	PD478A	01.00
CARD-READ/PUNCH	PD479A	01.00
OPTICAL MARK READER	PD480A	00.00

DOCUMENTATION

The following tables list currently available customer manuals for HP 3000 products. This list supersedes the list in the last issue of the COMMUNICATOR.

The most recent changes to the tables are indicated for easy reference. Prices are subject to change without notice.

Copies of manuals and updates 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 customer manuals.

Update packages are free of charge. If you require an update package complete the Update Order Form in the back of the COMMUNICATOR 3000 and mail the form to:

Software/Publications Distribution
5303 Stevens Creek Blvd.
Santa Clara, Ca 95050

Customers in the U.S. may also order directly by mail. Simply list the name and part number of the manual(s) you need on the Corporate Parts Center form supplied at the back of the COMMUNICATOR 3000.

A few words about documentation terms:

NEW — A new manual refers only to the first printing of a manual. When first printed, a manual is assigned a part number.

REVISED — A revised manual is a printing of an existing manual which incorporates new and/or changed information in its contents. For example, a manual is revised when an update package is incorporated into the manual: the manual gets a new print date and the update package disappears. Note that a revision to a manual effectively obsoletes the previous version of the manual.

UPDATE — An update package is a supplement to an existing manual which contains new and/or changed information. Updates are issued when information must get to customers, yet it is inappropriate to issue a revised manual. An update has no part number, it is automatically included when you order the manual with which it is associated.

HP 3000 SERIES II COMPUTER SYSTEM

SYSTEM MANUALS

Manual Title	Part Number	Price	Printed	Updated
Console Operator's Guide	30000-90013	$ 7.00	6/76	10/76
Error Messages and Recovery Manual	30000-90015	21.50	6/76	
General Information Manual	30000-90008	6.50	10/76	
HP 3000 CX to HP 300 0 Series II Program Conversion Guide	30000-90046	4.00	6/76	
Index to MPE Reference Documents	30000-90045	5.50	6/76	
MPE Commands Reference Manual	30000-90009	12.50	6/76	
MPE Intrinsics Reference Manual	30000-90010	17.00	2/77	
MPE Segmenter Reference Manual	30000-90011	4.00	2/77	
MPE Debug/Stack Dump Reference Manual	30000-90012	7.50	6/76	*9/76*
MPE System Utilities Reference Manual	30000-90044	5.00	3/77	
- Software Pocket Guide	30000-90049	3.50	1/77	
System Manager/System Supervisor Manual	30000-90014	10.00	2/77	*9/76*
Using the HP 3000: Guide for the Terminal User	03000-90121	6.00	6/75	

When a date in the "Updated" column is enclosed by asterisks, it means that the update has been incorporated in subsequent printings of the manual.

LANGUAGE MANUALS

Manual Title	Part Number	Price	Printed	Updated
APL Reference Manual	32105-90002	12.50	11/76	
APL Pocket Guide	32105-90003	1.50	11/76	
BASIC Interpreter Manual	30000-90026	$11.50	6/76	
BASIC Compiler Reference Manual	32103-90001	2.45	11/74	6/76
BASIC/3000 Pocket Guide	03000-90050	.75	9/74	
BASIC for Beginners	03000-90025	4.50	11/72	
COBOL Reference Manual	32213-90001	12.50	7/75	*6/76*
Cross Assembler for 2100 Computers Reference Manual	03000-90047	12.00	5/76	
FORTRAN Reference Manual	30000-90040	9.50	6/76	2/77
FORTRAN Pocket Guide	32102-90002	1.50	11/76	
RPG/3000 Compiler Reference Manual	32104-90001	22.00	2/77	
RPG Listing Analyzer	32104-90003	.20	2/77	
SPL Pocket Guide	32100-90001	2.00	11/76	
System Programming Language Reference Manual	30000-90024	17.50	9/76	2/77
System Programming Language Textbook	30000-90025	11.00	6/76	1/77

ADDITIONAL MANUALS

Manual Title	Part Number	Price	Printed	Updated
2780/3780 Emulator Reference Manual	30000-90047	8.50	6/76	1/77
Compiler Library Reference Manual	30000-90028	12.00	11/76	

Data Entry Library Mnl	30000-90050	6.50	6/76	
DS/3000 Reference Manual	32190-90001	13.50	3/77	
EDIT Reference Manual	03000-90012	6.25	8/75	*6/76*
FCOPY Reference Manual	03000-90064	5.00	6/76	*12/76*
HP 2894A Card Reader Punch Operating Manual	30119-90009	11.50	10/76	
HP 3000 Cross Loader for HP 2100 Computers	03000-90107	4.50	10/74	*6/76*
IMAGE Data Base Management Reference Manual	30000-90041	4.50	12/76	
Instruction Decoding Pocket Guide	30000-90057	1.00	6/76	
KSAM Reference Manual	30000-90079	13.00	1/77	
Line Printer Operating and Programming Manual	30209-90008	6.00	6/76	
Machine Instruction Set Reference Manual	30000-90022	7.00	6/76	
Programmable Controller Reference Manual	30000-90066	6.00	6/76	10/76
QUERY Reference Manual	30000-90042	6.50	6/76	
Real-Time Programmable Controller Reference	30000-90067	7.50	6/76	
Scientific Library Reference Manual	30000-90027	5.00	6/76	2/77
Site Preparation Manual	30000-90082	7.50	2/77	
Site Planning Workbook	30000-90086	10.00	2/77	
SORT Reference Manual	32214-90001	4.30	8/76	
Student Assignment System Reference Manual	32901-90001	15.50	7/75	8/76
Student Assignment System Technical Manual	32901-90005	10.50	7/75	
Student Information System Reference Manual	32900-90001	13.00	9/74	8/76

Student Information System Overview	32900-90002	3.00	9/74	
Student Information System Technical Mnl	32900-90005	15.00	3/75	
Systems Reference Manual	30000-90020	9.50	6/76	1/77
IBM System/3 to HP 3000 Conversion Guide	32104-90004	5.75	12/75	
Trace Reference Manual	03000-90015	5.00	6/76	

When a date in the "Updated" column is enclosed by asterisks, it means that the update has been incorporated in subsequent printings of the manual.

HP 3000 SERIES I COMPUTER SYSTEM

SYSTEM MANUALS

Manual Title	Part Number	Price	Printed	Updated
Console Operator's Guide	30000-90090	$11.00	4/77	
General Information Mnl	30000-90091	9.00	4/77	
MPE Intrinsics Reference Manual	30000-90087	20.00	4/77	
MPE Commands Reference	30000-90088	20.00	4/77	
MPE/3000 Operating System, System Utilities	32000-90008	2.05	10/75	
Software Pocket Guide	03000-90126	2.70	7/75	
System Manager/System Supervisor Manual	30000-90089	13.00	4/77	
Using the HP 3000: Guide for the Terminal User	03000-90121	6.00	6/75	

LANGUAGE MANUALS

Manual Title	Part Number	Price	Printed	Updated
BASIC Interpreter Reference Manual Pre-Series II	03000-90008	$ 9.75	7/75	
BASIC/3000 Pocket Guide	03000-90050	.75	9/74	
BASIC Compiler Reference	32103-90001	2.45	11/74	6/76
BASIC for Beginners	03000-90025	4.50	11/72	
COBOL Reference Manual	32213-90001	12.50	7/75	*6/76*
Cross Assembler for 2100 Computers	03000-90047	12.00	5/76	
FORTRAN Reference Manual	32102-90001	10.00	3/76	
RPG Compiler Reference and Application Manual	32104-90001	22.00	2/77	
RPG Listing Analyzer	32104-90003	.20	2/77	
System Programming Language Reference Manual	30000-90024	17.50	9/76	2/77
System Programming Language Textbook	30000-90025	11.00	6/76	1/77

ADDITIONAL MANUALS

Manual Title	Part Number	Price	Printed	Updated
2780/3780 Emulator Subsystem Reference Mnl	30130-90001	6.50	12/74	2/76
Compiler Library Reference Manual	03000-90009	11.50	2/76	
Data Entry Library Mnl	30000-90050	6.50	6/76	
EDIT Reference Manual	03000-90012	6.25	8/75	
FCOPY Reference Manual	03000-90064	5.00	6/76	*12/76*
HP 2894A Card Reader Punch Operating Manual	30119-90009	11.50	10/76	
HP 3000 Cross Loader for HP 2100 Computers	03000-90107	4.50	10/74	*6/76*

IBM 1130/1800 to HP 3000 FORTRAN Conversion Gd	36995-90013	4.70	2/75	5/75
IBM System/3 to HP 3000 Conversion Guide	32104-90004	5.75	12/75	
IMAGE Data Base Management Reference Manual	30000-90041	4.50	12/76	
Programmable Controller Reference Manual	30300-90002	12.50	4/76	1/77
QUERY Reference Manual	30000-90042	6.50	6/76	
Real-Time Programmable Controller Reference	30301-90002	7.75	2/75	7/76
Scientific Library Reference Manual	03000-90010	5.75	7/75	
Site Planning Workbook	30000-90100	5.00	4/77	
Site Preparation Manual	30000-90096	5.00	4/77	
SORT Reference Manual	32214-90001	4.30	8/76	
Student Assignment System Reference Manual	32901-90001	15.50	7/75	6/76
Student Assignment System Technical Manual	32901-90005	10.50	7/75	
Student Information System Reference Manual	32900-90001	13.00	9/74	6/76
Student Information System Overview	32900-90002	3.00	9/74	
Student Information System Technical Mnl	32900-90005	15.00	3/75	
Systems Reference Manual HP 3000 Computer	03000-90019	15.00	9/73	3/77
Trace Reference Manual	03000-90015	5.00	6/76	

When a date in the "Updated" column is enclosed by asterisks, it means that the update has been incorporated in subsequent printings of the manual.

BAUD LINE

NEW: HP 3000 CONTRIBUTED LIBRARY, VOLUME II

by Brenda Mapp
HP General Systems

A new volume of the HP 3000 Contributed Library is now available. It includes a software catalog and index, Volume II, Extended Documentation Package, Volume II, and a "STORE" magnetic tape containing 44 new programs and 22 revised programs from Volume I.

If you are a "site" member in the HP 3000 Users Group you will receive a special distribution of the HP 3000 Contributed Library which includes both Volume I and II plus special documentation.

If you do not have a "site" membership in the Users Group you may order the library by using the following numbers:

36995-10902	800 bpi Library	$77.00
36995-11902	1600 bpi Library	$77.00

This is a complete package which includes a magentic tape, the library catalog and Index for Volume II, and the Extended Documentation Package for Volume II.

Individual items are available by ordering the following part numbers:

36995-10002	800 bpi Magnetic Tape	$50.00
36995-11002	1600 bpi Magnetic Tape	$50.00
36995-90002	Index and Catalog, Vol. II	$ 7.50
36995-90902	Extended Documentation Package - Vol. II	$19.50

The Extended Documentation Package for Volume II consists of the following extended docmentation for the listed programs. You may order the entire package as part number 36995-90902, or you may use the following order numbers to obtain additional copies of individual extended documentation.

Contribution Number	Program Name	Order Number	Price
30	COBINTR	36995-90012	$ 1.00
40	COGO	*36995-90901	7.50
90	FORMSGN	36995-90014	.50
108	STAR	03000-90011	4.50
111	KONVERT	36995-90015	.50
116	IDEA	36999-90001	1.50
124	IDEAII	36995-90018	1.50
127	RPG/DEL	36995-90016	.50
130	CARDRDR	36995-90017	2.00

Please contact your local sales office to order the HP 3000 Contributed Library or use the Direct Mail order form - included in this publication. Please note: Only customers in the domestic U.S.A. may use this form. A check or money order must accompany each direct mail order which must include any applicable state and local taxes and an additional $1.50 to cover postage and handling.

*This is also the extended documentation for Volume I.

UPCOMING: CODE CONVERSION SOFTWARE CHANGE

by John Welsch
HP General Systems

There are currenty three different EBCDIC<--->ASCII character code conversion tables in use in the HP 3000 system. RPG and RJE use one, FCOPY uses another and the intrinsic CTRANSLATE uses yet another. The FCOPY and CTRANSLATE tables translate only 128 character codes and are not invertable (i.e. EBCDIC --->ASCII --->EBCDIC is not the identity transformation).

The RPG and RJE table translates the full 256 character codes and is invertable.

We plan to change the CTRANSLATE table on both HP 3000 Series I and II so that it is the same as RPG and RJE tables, then modify all subsystems (FCOPY, RPG, RJE) so that they use the CTRANSLATE intrinsic to do all character code conversions. So that you may prepare for this change, following is the code conversion table. The change will be made with the next Master Installation Tape (M.I.T.) after M.I.T. date code 1709.

ASCII-EBCDIC CODE CONVERSION

HOW TO USE THIS TABLE

- The table is sorted by character code, each code being represented by its decimal, octal, and hexadecimal equivalent.
- Each row of the table gives the ASCII and EBCDIC meaning of the character code, the ASCII↔EBCDIC conversion code, and the Hollerith representation (punched card code) for the ASCII character.

The following examples describe several ways of using the table:

Example 1: Suppose you want to determine the ASCII code for the $ character. Scan down the ASCII graphic column until you locate $, then look left on that row to find the character code – 36 (dec), 044 (oct), and 24 (hex). This is the code used by an ASCII device (terminal, printer, computer, etc.) to represent the $ character. Its Hollerith punched card code is 11-3-8.

Example 2: The character code 5B (hex) is the EBCDIC code for what character? Also, when 5B is converted to ASCII (for example, by FCOPY with the EBCDICIN option), what is the octal character code? First, locate 5B in the hex character code column and move right on that row to the EBCDIC graphic which is $. The next column to the right gives the conversion to ASCII, 044. As a check, find 044 (oct) in the character code column, look right to the ASCII graphic column and note that $ converted to EBCDIC is 133 (oct) which equals 5B (hex).

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
0	000	00	NUL	000	12-0-1-8-9	NUL	000
1	001	01	SOH	001	12-1-9	SOH	001
2	002	02	STX	002	12-2-9	STX	002
3	003	03	ETX	003	12-3-9	ETX	003
4	004	04	EOT	067	7-9	PF	234
5	005	05	ENQ	055	0-5-8-9	HT	011
6	006	06	ACK	056	0-6-8-9	LC	206
7	007	07	BEL	057	0-7-8-9	DEL	177
8	010	08	BS	026	11-6-9		227
9	011	09	HT	005	12-5-9		215
10	012	0A	LF	045	0-5-9	SMM	216
11	013	0B	VT	013	12-3-8-9	VT	013
12	014	0C	FF	014	12-4-8-9	FF	014
13	015	0D	CR	015	12-5-8-9	CR	015
14	016	0E	SQ	016	12-6-8-9	SO	016
15	017	0F	SI	017	12-7-8-9	SI	017
16	020	10	DLE	020	12-11-1-8-9	DLE	020
17	021	11	DC1	021	11-1-9	DC1	021
18	022	12	DC2	022	11-2-9	DC2	022
19	023	13	DC3	023	11-3-9	TM	023
20	024	14	DC4	074	4-8-9	RES	235
21	025	15	NAK	075	5-8-9	NL	205
22	026	16	SYN	062	2-9	BS	010
23	027	17	ETB	046	0-6-9	IL	207
24	030	18	CAN	030	11-8-9	CAN	030
25	031	19	EM	031	11-1-8-9	EM	031
26	032	1A	SUB	077	7-8-9	CC	222
27	033	1B	ESC	047	0-7-9	CU1	217
28	034	1C	FS	034	11-4-8-9	IFS	034
29	035	1D	GS	035	11-5-8-9	IGS	035
30	036	1E	RS	036	11-6-8-9	IRS	036
31	037	1F	US	037	11-7-8-9	IUS	037
32	040	20	SP	100	Blank	DS	200
33	041	21	!	117	12-7-8	SOS	201
34	042	22	"	177	7-8	FS	202
35	043	23	#	173	3-8		203
36	044	24	$	133	11-3-8	BYP	204
37	045	25	%	154	0-4-8	LF	012
38	046	26	&	120	12	ETB	027
39	047	27	'	175	5-8	ESC	033
40	050	28	(	115	12-5-8		210
41	051	29	)	135	11-5-8		211
42	052	2A	*	134	11-4-8	SM	212
43	053	2B	+	116	12-6-8	CU2	213
44	054	2C	,	153	0-3-8		214
45	055	2D	–	140	11	ENQ	005
46	056	2E	.	113	12-3-8	ACK	006
47	057	2F	/	141	0-1	BEL	007

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
48	060	30	0	360	0		220
49	061	31	1	361	1		221
50	062	32	2	362	2	SYN	026
51	063	33	3	363	3		223
52	064	34	4	364	4	PN	224
53	065	35	5	365	5	RS	225
54	066	36	6	366	6	UC	226
55	067	37	7	367	7	EOT	004
56	070	38	8	370	8		230
57	071	39	9	371	9		231
58	072	3A	:	172	2-8		232
59	073	3B	;	136	11-6-8	CU3	233
60	074	3C	<	114	12-4-8	DC4	024
61	075	3D	=	176	6-8	NAK	025
62	076	3E	>	156	0-6-8		236
63	077	3F	?	157	0-7-8	SUB	032
64	100	40	@	174	4-8	SP	040
65	101	41	A	301	12-1		240
66	102	42	B	302	12-2		241
67	103	43	C	303	12-3		242
68	104	44	D	304	12-4		243
69	105	45	E	305	12-5		244
70	106	46	F	306	12-6		245
71	107	47	G	307	12-7		246
72	110	48	H	310	12-8		247
73	111	49	I	311	12-9		250
74	112	4A	J	321	11-1	¢	133
75	113	4B	K	322	11-2	.	056
76	114	4C	L	323	11-3	<	074
77	115	4D	M	324	11-4	(	050
78	116	4E	N	325	11-5	+	053
79	117	4F	O	326	11-6	|	041
80	120	50	P	327	11-7	&	046
81	121	51	Q	330	11-8		251
82	122	52	R	331	11-9		252
83	123	53	S	342	0-2		253
84	124	54	T	343	0-3		254
85	125	55	U	344	0-4		255
86	126	56	V	345	0-5		256
87	127	57	W	346	0-6		257
88	130	58	X	347	0-7		260
89	131	59	Y	350	0-8		261
90	132	5A	Z	351	0-9	!	135
91	133	5B	[	112	12-2-8	$	044
92	134	5C	\	340	0-2-8	*	052
93	135	5D	]	132	11-2-8	)	051
94	136	5E	^	137	11-7-8	;	073
95	137	5F	_	155	0-5-8	¬	136

ASCII-EBCDIC CODE CONVERSION (CONTINUED)

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
96	140	60	`	171	1-8	-	055
97	141	61	a	201	12-0-1	/	057
98	142	62	b	202	12-0-2		262
99	143	63	c	203	12-0-3		263
100	144	64	d	204	12-0-4		264
101	145	65	e	205	12-0-5		265
102	146	66	f	206	12-0-6		266
103	147	67	g	207	12-0-7		267
104	150	68	h	210	12-0-8		270
105	151	69	i	211	12-0-9		271
106	152	6A	j	221	12-11-1	¦	174
107	153	6B	k	222	12-11-2	,	054
108	154	6C	l	223	12-11-3	%	045
109	155	6D	m	224	12-11-4	_	137
110	156	6E	n	225	12-11-5	>	076
111	157	6F	o	226	12-11-6	?	077
112	160	70	p	227	12-11-7		272
113	161	71	q	230	12-11-8		273
114	162	72	r	231	12-11-9		274
115	163	73	s	242	11-0-2		275
116	164	74	t	243	11-0-3		276
117	165	75	u	244	11-0-4		277
118	166	76	v	245	11-0-5		300
119	167	77	w	246	11-0-6		301
120	170	78	x	247	11-0-7		302
121	171	79	y	250	11-0-8	`	140
122	172	7A	z	251	11-0-9	:	072
123	173	7B	{	300	12-0	#	043
124	174	7C	¦	152	12-11	@	100
125	175	7D	}	320	11-0	'	047
126	176	7E	~	241	11-0-1	=	075
127	177	7F	DEL	007	12-7-9	"	042
128	200	80		040	11-0-1-8-9		303
129	201	81		041	0-1-9	a	141
130	202	82		042	0-2-9	b	142
131	203	83		043	0-3-9	c	143
132	204	84		044	0-4-9	d	144
133	205	85		025	11-5-9	e	145
134	206	86		006	12-6-9	f	146
135	207	87		027	11-7-9	g	147
136	210	88		050	0-8-9	h	150
137	211	89		051	0-1-8-9	i	151
138	212	8A		052	0-2-8-9		304
139	213	8B		053	0-3-8-9		305
140	214	8C		054	0-4-8-9		306
141	215	8D		011	12-1-8-9		307
142	216	8E		012	12-2-8-9		310
143	217	8F		033	11-3-8-9		311
144	220	90		060	12-11-0-1-8-9		312
145	221	91		061	1-9	j	152
146	222	92		032	11-2-8-9	k	153
147	223	93		063	3-9	l	154
148	224	94		064	4-9	m	155
149	225	95		065	5-9	n	156
150	226	96		066	6-9	o	157
151	227	97		010	12-8-9	p	160
152	230	98		070	8-9	q	161
153	231	99		071	1-8-9	r	162
154	232	9A		072	2-8-9		313
155	233	9B		073	3-8-9		314
156	234	9C		004	12-4-9		315
157	235	9D		024	11-4-9		316
158	236	9E		076	6-8-9		317
159	237	9F		341	11-0-1-9		320
160	240	A0		101	12-0-1-9		321
161	241	A1		102	12-0-2-9	~	176
162	242	A2		103	12-0-3-9	s	163
163	243	A3		104	12-0-4-9	t	164
164	244	A4		105	12-0-5-9	u	165
165	245	A5		106	12-0-6-9	v	166
166	246	A6		107	12-0-7-9	w	167
167	247	A7		110	12-0-8-9	x	170
168	250	A8		111	12-1-8	y	171
169	251	A9		121	12-11-1-9	z	172
170	252	AA		122	12-11-2-9		322
171	253	AB		123	12-11-3-9		323
172	254	AC		124	12-11-4-9		324
173	255	AD		125	12-11-5-9		325
174	256	AE		126	12-11-6-9		326
175	257	AF		127	12-11-7-9		327

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
176	260	B0		130	12-11-8-9		330
177	261	B1		131	11-1-8		331
178	262	B2		142	11-0-2-9		332
179	263	B3		143	11-0-3-9		333
180	264	B4		144	11-0-4-9		334
181	265	B5		145	11-0-5-9		335
182	266	B6		146	11-0-6-9		336
183	267	B7		147	11-0-7-9		337
184	270	B8		150	11-0-8-9		340
185	271	B9		151	0-1-8		341
186	272	BA		160	12-11-0		342
187	273	BB		161	12-11-0-1-9		343
188	274	BC		162	12-11-0-2-9		344
189	275	BD		163	12-11-0-3-9		345
190	276	BE		164	12-11-0-4-9		346
191	277	BF		165	12-11-0-5-9		347
192	300	C0		166	12-11-0-6-9	{	173
193	301	C1		167	12-11-0-7-9	A	101
194	302	C2		170	12-11-0-8-9	B	102
195	303	C3		200	12-0-1-8	C	103
196	304	C4		212	12-0-2-8	D	104
197	305	C5		213	12-0-3-8	E	105
198	306	C6		214	12-0-4-8	F	106
199	307	C7		215	12-0-5-8	G	107
200	310	C8		216	12-0-6-8	H	110
201	311	C9		217	12-0-7-8	I	111
202	312	CA		220	12-11-1-8		350
203	313	CB		232	12-11-2-8		351
204	314	CC		233	12-11-3-8	⌠	352
205	315	CD		234	12-11-4-8		353
206	316	CE		235	12-11-5-8	⌡	354
207	317	CF		236	12-11-6-8		355
208	320	D0		237	12-11-7-8	}	175
209	321	D1		240	11-0-1-8	J	112
210	322	D2		252	11-0-2-8	K	113
211	323	D3		253	11-0-3-8	L	114
212	324	D4		254	11-0-4-8	M	115
213	325	D5		255	11-0-5-8	N	116
214	326	D6		256	11-0-6-8	O	117
215	327	D7		257	11-0-7-8	P	120
216	330	D8		260	12-11-0-1-8	Q	121
217	331	D9		261	12-11-0-1	R	122
218	332	DA		262	12-11-0-2		356
219	333	DB		263	12-11-0-3		357
220	334	DC		264	12-11-0-4		360
221	335	DD		265	12-11-0-5		361
222	336	DE		266	12-11-0-6		362
223	337	DF		267	12-11-0-7		363
224	340	E0		270	12-11-0-8	\	134
225	341	E1		271	12-11-0-9		237
226	342	E2		272	12-11-0-2-8	S	123
227	343	E3		273	12-11-0-3-8	T	124
228	344	E4		274	12-11-0-4-8	U	125
229	345	E5		275	12-11-0-5-8	V	126
230	346	E6		276	12-11-0-6-8	W	127
231	347	E7		277	12-11-0-7-8	X	130
232	350	E8		312	12-0-2-8-9	Y	131
233	351	E9		313	12-0-3-8-9	Z	132
234	352	EA		314	12-0-4-8-9		364
235	353	EB		315	12-0-5-8-9		365
236	354	EC		316	12-0-6-8-9	⌐	366
237	355	ED		317	12-0-7-8-9		367
238	356	EE		332	12-11-2-8-9		370
239	357	EF		333	12-11-3-8-9		371
240	360	F0		334	12-11-4-8-9	0	060
241	361	F1		335	12-11-5-8-9	1	061
242	362	F2		336	12-11-6-8-9	2	062
243	363	F3		337	12-11-7-8-9	3	063
244	364	F4		352	11-0-2-8-9	4	064
245	365	F5		353	11-0-3-8-9	5	065
246	366	F6		354	11-0-4-8-9	6	066
247	367	F7		355	11-0-5-8-9	7	067
248	370	F8		356	11-0-6-8-9	8	070
249	371	F9		357	11-0-7-8-9	9	071
250	372	FA		372	12-11-0-2-8-9	¦	372
251	373	FB		373	12-11-0-3-8-9		373
252	374	FC		374	12-11-0-4-8-9		374
253	375	FD		375	12-11-0-5-8-9		375
254	376	FE		376	12-11-0-6-8-9		376
255	377	FF		377	12-11-0-7-8-9	EO	377

NOTE: This order form cannot be used for ordering manuals. It is for replacement pages only.

HEWLETT PACKARD

SOFTWARE/PUBLICATIONS DISTRIBUTION
ORDER FORM

UPDATES TO 3000 AND 2000
LEVEL MANUALS ONLY

SHIP TO:

NAME _____

COMPANY _____

STREET _____

CITY _____ STATE _____ ZIP CODE _____

MANUAL NAME	PART NUMBER	QUANTITY

When completed, please mail this form to:

There is no charge for manual updates.

HEWLETT-PACKARD
SOFTWARE/PUBLICATIONS DISTRIBUTION
5303 Stevens Creek Blvd.
Santa Clara, CA 95050

HEWLETT-PACKARD
COMPUTER SYSTEMS COMMUNICATOR ORDER FORM

Please Print:

Name _____ **Title** _____

Company _____

Street _____

City _____ **State** _____ **Zip Code** _____

Country _____

☐ HP Employee Account Number _____ Location Code _____

☐ **DIRECT SUBSCRIPTION**

Part No.	Description	Qty	List Price	Extended Dollars	Total Dollars
5951-6111	COMMUNICATOR 1000	_____	$48.00	_____	
	(if quantity is greater than 1 discount is 40%)			_____	
	TOTAL DOLLARS for 5951-6111				_____
5951-6112	COMMUNICATOR 2000	_____	25.00	_____	
	(if quantity is greater than 1 discount is 40%)			_____	
	TOTAL DOLLARS for 5951-6112				_____
5951-6113	COMMUNICATOR 3000	_____	48.00	_____	
	(if quantity is greater than 1 discount is 40%)			_____	
	TOTAL DOLLARS for 5951-6113				_____

☐ **BACK ISSUE ORDER FORM (cash only in U.S. dollars)**
(subject to availability)

Part No.	Description	Issue No.	Qty	List Price	Extended Dollars	Total Dollars
5951-6111	COMMUNICATOR 1000	_____	_____	$10.00	_____	
		_____	_____	10.00	_____	
		_____	_____	10.00	_____	
	TOTAL DOLLARS					_____
5951-6112	COMMUNICATOR 2000	_____	_____	$ 5.00	_____	
		_____	_____	5.00	_____	
		_____	_____	5.00	_____	
	TOTAL DOLLARS					_____
5951-6113	COMMUNICATOR 3000	_____	_____	$10.00	_____	
		_____	_____	10.00	_____	
		_____	_____	10.00	_____	
	TOTAL DOLLARS					_____
TOTAL ORDER DOLLAR AMOUNT						_____

☐ **SERVICE CONTRACT CUSTOMERS**

You will receive one copy of either COMMUNICATOR 1000, 2000, or 3000 as part of your contract. Indicate additional copies below and have your local office forward. Billing will be included in normal contract invoices.

Number of additional copies _____

FOR HP USE ONLY

CONTRACT KEY

5951-6111 Number of additional copies _____
5951-6112 Number of additional copies _____
5951-6113 Number of additional copies _____

Approved _____

HEWLETT-PACKARD
COMMUNICATOR SUBSCRIPTION AND ORDER INFORMATION

The Computer Systems COMMUNICATORS are bi-monthly systems support publications available from Hewlett-Packard on an annual (6 issues) subscription.

The following instructions are for customers who do not have Software Service Contracts.

1. Complete name and address portion of order form.
2. For new direct subscriptions (see sample below):
   a. Indicate which COMMUNICATOR publication(s) you wish to receive.
   b. Enter number of copies per issue under Qty column.
   c. Extend dollars (quantity x list price) in Extended Dollars column.
   d. Enter discount dollars on line under Extended Dollars. (If quantity is greater than 1 you are entitled to a 40% discount.*)
   e. Enter Total Dollars (subtract discount dollars from Extended List Price dollars).

   **To qualify for discount all copies of publications must be mailed to same name and address and ordered at the same time.*

SAMPLE

☒ **DIRECT SUBSCRIPTION**

Part No.	Description	Qty	List Price	Extended Dollars	Total Dollars
5951-6111	COMMUNICATOR 1000	3	$48.00	$144.00	
	(if quantity is greater than 1 discount is 40%)			57.60	
	TOTAL DOLLARS for 5951-6111				$86.40

3. To order back issues (see sample below):
   a. Indicate which publication you are ordering.
   b. Indicate which issue number you want.
   c. Enter number of copies per issue.
   d. Extend dollars for each issue.
   e. Enter total dollars for back issues ordered.

   All orders for back issues of the COMMUNICATORS are cash only orders (U.S. dollars only) and are subject to availability.

SAMPLE

☒ **BACK ISSUE ORDER FORM (cash only in U.S. dollars)**
(subject to availability)

Part No.	Description	Issue No.	Qty	List Price	Extended Dollars	Total Dollars
5951-6111	COMMUNICATOR 1000	XX	1	$10.00	$10.00	
		XX	2	10.00	20.00	
				10.00		
	TOTAL DOLLARS					$30.00

4. Domestic Customers: Mail the order form with your U.S. Company Purchase Order or check (payable to Hewlett-Packard Co.) to:

   HEWLETT-PACKARD COMPANY
   Computer Systems COMMUNICATOR
   P.O. Box 61809
   Sunnyvale, CA 94088
   U.S.A.

5. International Customers: Order by part number through your local Hewlett-Packard Sales Office.

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.

Computer Systems **Communicator**
Subscription Service Manager
Hewlett-Packard Company
Mail Order Department
P.O. Drawer No. 20
Mountain View, California 94043
U.S.A. June 1977

Address Correction Requested
Forwarding and Return Postage Guaranteed

Printed in U.S.A. 6/77
Part No. 5951-6113