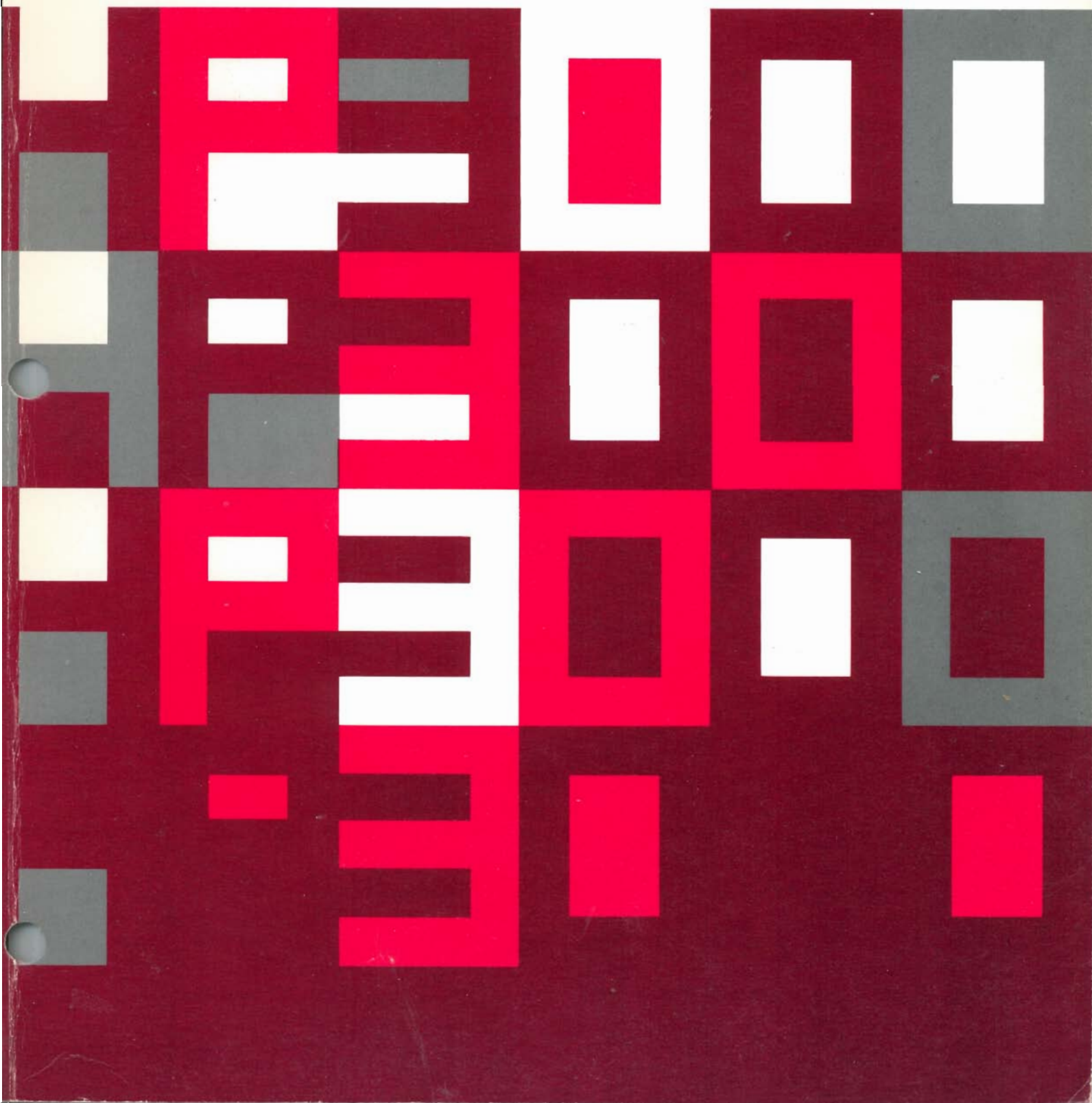
MPE IV 2131

HEWLETT PACKARD

COMMUNICATOR

ISSUE NUMBER 27



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Table of Contents

Editor's Note 3
Introducing the MPE IV D-IT Release 4
File System Enhancements for the "D" Installation Tape 8
Changes in Stack Size Requirements for D-IT 10
Automatic Volume Recognition of Tapes 11
Series III Software Update 12
Series 30/33/44 Software Update 18
Computer Data Better Presented at Lower Cost by
New Laser Printing System 23
The 2680A Laser Printing System from the
Programmer's Perspective 26
Introducing IDS/3000 and IFS/3000 Design and Formatting
Software for the 2680 Laser Printing System 34
Using IFS/3000 Intrinsics 38
Using Environment Files 46
The HP 2680A Translator 48
2680A Software Configuration 49
D-IT Data Communication Requirements 49
Spooler Powerfail Recovery Procedures for the
2680A Laser Page Printer 50
MPE Spooler Changes for the 2680A 51
Enhanced Laser Printer Support with DSN/MRJE and DSN/RJE 53
MRJE and RJE Performance on the HP 2680A Laser
Printing System 54
Introducing the On-Line Performance Tool/3000
Performance Measurement Package 74
High-Speed Half-Inch Tape Drive for HP Computers 82
7976A Tape Drive Performance and Tape Requirements 84
New On-Line Diagnostics for HP 2680A and HP 7976A 86
New Subtypes for Tape Drives 88
30341A HP-IB Interface Module for Series III 89
Series III System Startup with an HP-IB Interface 89
Using Labeled Tapes 90
COBOL vs COBOLII - Use of "PERFORM" and "GO TO"
Statements 101
Calling SPL from Other Languages on the HP 3000 106
Correspondence-Quality Printer for Hewlett-Packard
Computer Systems 112
Documentation 113
New Manuals 116
New Editions 118
Updates 119
Catalog 121

Editor's Note

This issue is "chock-full" of information about the new software contained in the "D" IT release, as well as information on two new peripherals, the HP 2680A Laser Page Printing System, and the HP 7976A High Speed Magnetic Tape Subsystem.

The first article, "Introducing the MPE IV D-IT Release", gives an overview of the software changes and the new products, and the following articles give more detailed information about the exciting changes to be found in the 2131 release.

New software for the 2680A is covered in "Introducing IDS/3000 and IFS/3000" beginning on page 34, and "Using IFS/3000 Intrinsics" (page 38) covers the new intrinsics developed for use with the 2680A.

The article on page 26, "The 2680A Laser Printing System from the Programmer's Perspective" gives a different outlook on the new Laser Printer. Environment Files, the 2680A Translator, and software configuration are covered in other 2680-related articles.

The new On-Line Performance Measurement Tool (OPT/3000) is explained in detail in an article beginning on page 74.

The 7976A Tape Drive is covered in four articles, beginning on page 82. Other articles discuss the new HP-IB Interface Module for Series III systems, (page 89), and Labelled Tapes, (page 90). The article "COBOL vs COBOLII", "Calling SPL From Other Languages", and a reprint of a press release on the new 2601A Printer complete this issue.

We hope you will find the COMMUNICATOR both informative and interesting, and invite you to submit your comments by writing to the Editor at the address listed below.

Editor
Communicator 3000
HP Computer Systems Division
19447 Pruneridge Avenue
Cupertino, Ca. 95014

Introducing the MPE IV D-IT Release

by Stephanie Littell and Greg Mihran, Computer Systems Division

The latest release (D Installation Tape, Revision 2131) of MPE IV brings the support of two new powerful peripherals to your HP 3000 computer system: the HP 2680A Laser Printing System and the HP 7976A Magnetic Tape Drive Subsystem. Additionally, the D-IT provides software to support the new HP 30341A HP-IB Interface Module which allows connection of these new HP-IB peripherals to HP 3000 Series III computer systems. As well as providing software for the new peripherals and interface modules, this release offers an enhanced Tape Labels capability for all of the HP 3000 systems. With this latest release of MPE IV, these new system capabilities and additional hardware reaffirm Hewlett-Packard's continuing commitment to the HP 3000 installed base. These new hardware and software capabilities are detailed in other articles in this issue of the COMMUNICATOR. A general overview is provided below.

The HP 2680A Laser Printing System

The MPE file system has been enhanced to allow for the increased flexibility needed to support the HP 2680A Laser Printing System. The HP 2680A utilizes a laser scanned, electrophotographic process to print 45 pages per minute using plain 8 1/2 by 11 inch paper. It is now possible to print spoolfile data on an HP 2680A using electronic forms, logos, character sets, and page orientations specified ahead of time via an HP 3000 terminal. With other HP printers, special forms must be supplied to achieve the same result. The MPE file system has undergone some changes to allow for this increased flexibility in forms handling.

Character set and page orientation information, as well as forms and logo information, can be sent to the HP 2680A from the HP 3000 host system along with the spoolfile data to be printed. All of the special information sent to the HP 2680A is contained in an "environment file". This file describes the "environment" that the spoolfile data will be printed in, and thus influences how the printed output will vary in appearance. The FILE command and the FOPEN intrinsic are used to relate a specific environment with a specific spoolfile.

At the Command Interpreter level, the environment specification is accomplished by:

```
:FILE <filename>;DEV=<2680devclassname>;ENV=<environmentfilename>
```

The device class name should be associated with an HP 2680A. The environment file specified should be one supplied by HP with the printer or one created by using the new HP 2680A software subsystems: Interactive Design System/3000 and Interactive Formatting System/3000. These new software subsystems are described elsewhere in this COMMUNICATOR issue.

The programmatic means of specifying an environment file with a spoolfile is done by utilizing the FOPEN intrinsic. The environment file here is indirectly referenced through the FOPEN device specification:

```
num:=FOPEN(filename, foptions, aoptions, recsize, device, ...);
```

In another section of the same program, the byte array "device" must be defined as a combination of the device class name, the environment file name, and a carriage return. This is fully documented in the current update to the MPE Intrinsics Manual.

A new intrinsic has been developed to take advantage of the HP 2680A's capabilities. It is similar to the FCONTROL intrinsic but allows for more programmatic options. The format is as follows:

```
FDEVICECONTROL (Filenum, Target, Tcount, Cntlcode, P1, P2,
Errnum);
```

New return parameters in the FCHECK AND FFILEINFO intrinsics have been included to allow for the new intrinsic FDEVICECONTROL and for the changes to FOPEN.

The spooler has been enhanced to accommodate the new HP 2680A capabilities as well. The RESUMESPOOL console operator command contains a backwards/forwards spacing option as follows:

```
:RESUMESPOOL  ldn  [;BACK     {nnn PAGES}]
                   [;FORWARD  {nnn FILES}]
                   [;BEGINNING           ]
```

This option allows for print restarting on page boundaries rather than on file boundaries. Restarting is possible up to 256 pages preceding or following the current page, since extents are not purged from a 2680A spoolfile.

Note that a FILE is defined in a spoolfile as wherever an FOPEN occurs. Use of the FILES parameter is not allowed on the 2680A, since the entire spoolfile is purged after printing.

The HP 7976A Magnetic Tape Drive Subsystem

The MPE file system has been enhanced to allow for the addition of the new high speed, high bit density HP 7976A Tape Drive. This tape drive has the ability to read and write tapes at two different densities: 6250 BPI and 1600 BPI.

The density specification can be handled either programmatically or through the Command Interpreter. If the Command Interpreter is used, it can be specified as follows:

```
:FILE <filename>; DEV=<7976devclassname>; DEN=6250
```

The device class name should be associated with an HP 7976A. The density function selected will only be applied when the tape drive is on-line, at the load point, and about to be written.

The programmatic version of this function is accomplished by using the FOPEN intrinsic. An optional keyword parameter, ";DEN=", has been added to the device parameter of that intrinsic. This is analogous to FOPEN's treatment of the HP 2680A environment files described earlier. Both were designed to allow more flexibility in the FOPEN intrinsic and are further detailed in the MPE Intrinsics Manual.

The HP 30341A HP-IB Interface Module

The D-IT also provides software to support the new HP 30341A HP-IB Interface Module. This new module will permit users to connect the HP 2680A Laser Printer and the HP 7976A Magnetic Tape Drive to the HP 3000 Series III. The only modifications visible to MPE users are the startup procedures and configuration restrictions necessary to support the HP 30341A HP-IB Interface Module on the Series III. All other software changes required to allow users to access these peripherals on a Series III have been made transparent to the user. The only physical difference users will detect is the additional Series III-style bay sitting to the right side of the existing system bay which houses the HP-IB Interface Module.

The Newly Enhanced HP 3000 Tape Labels Capability

The Tape Labels capability on the HP 3000 has been completely re-written to offer the user better performance and ease of use. Labeled tapes are used to provide a permanent identification for tape reels or volumes. Tape Labels provide labels for files which extend over more than one volume, labels for multiple files on a volume, or retrieval of files by file name. Tape Labels also provide additional file security to protect against acci-

dental erasure or unauthorized access to files. The software allows the user to read and write ANSI standard labels and read IBM standard labels. Additionally, when a tape is mounted and put on-line, the tape label is read and the volume name and volume set are reported to the operator's console.

File System Enhancements for the 'D' Installation Tape

by Mike Paivinen, Computer Systems Division

The "D" IT File System includes the following enhancements:

BUFFER SPACE

Maximum total buffer space available for each file has been increased from 16KB to 28KB. Files with block sizes in the range of 8193 to 14000 words may now be accessed with a single buffer, and files with block sizes in the range of 4097 to 7000 words may be accessed with two buffers.

FILE COMMAND

New optional keyword parameters, DENsity and ENVironment have been added to the :FILE command. These allow the selection of density on the 7976A Tape Drive and specification of the environment file when using the 2680A Laser Printer.

```
;DEN= [1600]
      [6250]

;ENV= [environmentfilename]
```

Also, the <volid> parameter of the LABEL keyword, which is used for specifying the volume ID of a labelled tape, has been expanded as follows:

```
                <volume id>
volid ::=          or
          "<volume id>"
```

In the first form, <volume id> may be up to six (6) alphanumeric characters. To include special characters in the volume id, the second form must be used. With this form, <volume id> may be up to six (6) printable characters, excluding comma (,) and semicolon (;). The quotation mark (") may be included as a character in the volume ID by doubling it. Note that lower case characters are not upshifted.

FOPEN

The FOPEN Intrinsic has been modified to allow inclusion of the ENV and DEN parameters in the DEVICE ARRAY parameter. Also, the label portion of the FORMMSG array has been modified as follows:

1) <volumeid> may be up to six (6) printable characters, excluding comma (,) and semicolon (;). Unlike the :FILE command, enclosing quotation marks (") are not needed to indlude special characters in the volume ID.

2) <expdate>, the expiration date for the file, may be expressed as a positive offset from the current date. The syntax is:

expdate ::= + <positive integer>

For example, if <expdate> is "+1", the tape file expires the next day.

FFILEINFO

A new item number, 46, has been added to FFILEINFO to return the density of a tape file on a 7976A Tape Drive. The Intrinsic returns a density of zero (0) if the file does not reside on a 7976A. Also, a new item number, 45, has been added to return the file name from the HDR1 label of a labelled tape file.

Changes in Stack Size Requirements for D-IT

by Stephanie Littell, Computer Systems Division

The FOPEN intrinsic has been enhanced to allow for the ability to link a specified environment file with a spoolfile. The referenced environment file may be either remote or local to the user system. This feature has increased the stack size requirements for those FOPEN calls specifying an environment file. This can be seen in the following table:

FOPEN conditions	Approx. stack size (words)
No environment file (Same as C-IT)	893
Default local environment file, directed to a local 2680	1040
Specified local environment file, directed to a local 2680	2048
Specified remote environment file, directed to a local 2680	1997
Specified local environment file, directed to a remote 2680	2048
Specified remote environment file, directed to a remote 2680	1969

All other file system intrinsics use the same amount of stack space on D-IT as they did on C-IT.

With the FOPEN increases in stack size, it may be necessary for you to increase the initial stack size and/or the maxdata value for some of your programs. Another alternative is to reconfigure either the maximum or standard stack size configured into the system as a result of the above changes.

Automatic Volume Recognition of Tapes

by Bob Mead, Computer Systems Division

As a result of some changes to the tape label code in MPE, the procedures which perform automatic volume recognition (AVR) of tapes now require an additional 68 words of stack space. Normally, these procedures execute on the stack of the system process, PROGEN. However, if the REPLY for a tape request is made before the tape is placed on-line, thus assigning the tape drive to the process requesting the tape, the AVR procedures will execute on the user process stack (in much the same manner as MPE intrinsics execute on the user process stack). If this situation occurs, and the user process stack is close to the MAXDATA value specified for the program, it is likely that the program will abort with a stack overflow.

If this occurs, three solutions are possible:

(1) PREP the program with a larger MAXDATA

(2) Run the program with a larger MAXDATA specified on the :RUN command

(3) When the program requests a tape, mount the tape and place it on-line BEFORE replying to the tape request.

Solution 1 will permanently prevent the situation from occurring (for the program which aborts), whereas solution 2 must be used every time the program in question is executed. Solution 3 is recommended if it is not practical, or possible, to use either solution 1 or solution 2.

There is no effect on programs which do not use tapes.

Series II/III Software Update

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES II/III
****

CONTENTS OF INSTALLATION TAPE DATE CODE '2131'

PRODUCTS WITH ASTERISKS ARE THE PRODUCT(S) UPDATED/CHANGED
BY THIS M.I.T..

****

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE
*MPE	32002C	00.02	2131
SEGMENTER	32050A	01.03	2052
SPL	32100A	08.01	2052
BASIC	32101B	00.14	2052
FORTRAN	32102B	01.05	2052
BASIC COMPILER	32103B	00.14	2052
RPG	32104A	04.07	2052
APL/3000	32105A	01.04	2028
BUILDINT	32150A	03.01	1623
*DS/3000	32190A	03.02	2131
*MRJE	32192A	01.03	2131
MTS	32193A	03.05	2052
EDITOR	32201A	07.09	2052
SCIENTIFIC LIBRARY	32205B	00.04	1906
DEL/3000	32206A	01.10	2011
KSAM/3000	32208A	03.03	2101
*V/3000	32209B	02.00	2131
COMPILER LIBRARY	32211D	01.02	2052
FCOPY	32212A	03.13	2052
COBOL	32213C	02.07	2052
*SORT/MERGE	32214C	02.06	2131
*IMAGE	32215B	03.01	2131
QUERY	32216A	04.03	2011
TRACE	32222A	03.03	1814
XA2100	32223A	01.04	2011
XL2100	32226A	02.00	1636
IML	32229A	00.01	2052
*DIAGNOSTICS	32230A	01.05	2131
COBOL LIBRARY	32232A	00.03	2052
COBOLII	32233A	00.03	2052

CONTENTS OF INSTALLATION TAPE DATE CODE '2131'(Continued)

PRODUCTS WITH ASTERISKS ARE THE PRODUCT(S) UPDATED/CHANGED
BY THIS M.I.T..

****

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE
*OPT/3000	32238A	00.02	2131
DATACAPTURE PROCEDURES	32243A	01.00	2028
*DSG	32250A	00.01	2131
*DIAGNOSTICS	32340A	2132	2131
*DIAGNOSTICS	32341A	2126	2131
*TDP/3000	36578A	01.00	2131
*IFS/3000	36580A	00.01	2131
*IDS/3000	36581A	00.01	2131
PROG CONTROLLER 30300B/30361B-BCS	30361B	00.00	1621
PROG CONTROLLER 30301B/30361B-1-RTE	30361B-1	00.02	1701
RJE 2780/3780	30130E	01.00	2052
CALCOMP PLOTTER	30126A	00.01	1640

MPE HP32002C.00.02

I. MPE 32002C.00.02

A. MODULES MODIFIED C.00.02

	MODULE		CHANGE HISTORY												
	NAME	NO	A.01.XX			B.00.XX			B.01.XX				C.00.XX			
			1	2	MR	0	1	2	0	1	2	3	0	1	2	3
	INITIAL	0	X		X	X	X	X	X	X	X		X		X	
	SYSDUMP	1	X		X	X	X	X	X	X	X		X		X	
	SEGPROC	2								X			X			
	SEGDVR	3								X			X			
*	DISPATCH	4				X			X	X						
	LOAD	5				X	X	X	X	X			X			
	UCOP	7				X	X	X		X			X		X	
	DEVREC	8				X	X	X	X	X			X		X	
	PROGEN	9	X		X	X	X	X	X	X	X		X		X	
	ININ	10	X	X		X		X	X	X	X		X	X	X	
	MEMLOGP	11				X			X		X				X	
	LOG	12		X		X				X	X		X			

(Continued)

	MODULE		CHANGE HISTORY												
	NAME	NO	A.01.XX			B.00.XX			B.01.XX				C.00.XX			
			1	2	MR	0	1	2	0	1	2	3	0	1	2	3
	IOPTRD0	13				X					X					
	IOPTPN0	14		X		X					X		X			
	IOPLOT0	15				X					X				X	
	IOMDISC0	16									X					
	IOFDISC0	17									X					
	IOTAPE0	18	X	X		X		X	X	X	X				X	
	IOLPRT0	19				X		X	X	X	X					
	IOCDRD0	20				X		X	X	X	X					
	HIOLPRT2	21														
	IOTERM0	22	X	X		X		X	X	X	X		X			
	IOPRPN0	24		X		X		X	X		X					
	HIOPPRT0	23													X	
	IOREM0	25									X					
	HIOTERM0	26									X		X	X		
	IOMDISC1	27		X		X				X	X		X			
	PFAIL	30						X					X	X		
	PVPROC	31				X	X			X	X		X			
	VINIT	32				X	X	X		X	X		X			
	SDFLOAD	33													X	
	SDFGEN	34									X					
	HIOTAPE0	35									X		X		X	
	HIOLPRT0	36									X		X			
	HIOMDSC1	37									X		X		X	
	HIOLPRT1	38									X					
	HIOFLOP0	39									X		X		X	
	MAKECAT	40				X		X	X	X	X		X			
	HIOTAPE1	41													X	
	FILEACC	50	X	X	X	X	X	X	X	X	X		X		X	
	COMM'INT	51	X	X	X	X	X	X	X	X	X		X		X	
	STORE/RESTORE	52	X			X		X	X	X			X		X	
	DIRC	53				X	X	X	X	X	X					
	ALLOCATE	54	X	X		X	X	X	X	X	X		X		X	
*	SOFTRES	56	X	X		X	X	X	X	X	X					
	HARDRES	55								X	X		X	X	X	
*	MMDISKR	57	X	X		X	X	X	X	X	X					
	ABORTDUMP	58		X		X		X		X			X			
	MESSAGE	59				X	X		X	X	X		X			
	PROCSEG	60	X	X		X		X	X	X	X		X			
	NRIO	62	X	X		X		X	X	X	X		X		X	
	PCREATE	63		X		X		X		X	X		X			
	MORGUE	64	X	X		X		X	X	X	X		X	X	X	
	BIPC	65											X			
	IPC	66											X			
*	DATASEG	67	X	X		X	X	X	X	X						
	CHECKER	69				X	X	X		X			X			

(Continued)

MODULE		CHANGE HISTORY													
NAME	NO	A.01.XX			B.00.XX			B.01.XX				C.00.XX			
		1	2	MR	0	1	2	0	1	2	3	0	1	2	3
UTILITY	70	X	X		X			X	X	X		X			
SEGUTIL	71								X			X			
LOADER1	72		X		X	X	X	X	X	X		X			
RINS	73					X	X	X	X			X			
JOBTABLE	74	X	X		X	X	X	X	X					X	
DEBUG	75				X		X	X	X	X		X			
NURSERY	76				X	X	X	X	X	X					
FIRMWARESIM	78				X							X			
SPOOLING	79	X	X		X		X	X	X	X		X	X	X	
SPOOLCOMS	80	X	X		X		X	X	X	X		X		X	
PVSYS	81				X	X	X		X	X		X			
UDC	82				X		X	X	X	X		X			
USER	83				X				X	X		X			
HELPUSER	84				X				X	X		X			
OPCOMMND	85							X	X	X		X	X	X	
LABSEG	86				X	X	X	X	X					X	
SDISC	87				X	X	X	X	X			X			
MEASIO	88							X	X	X		X			
LOGSEG0	90							X	X	X		X	X		
LOGSEG1	91							X	X	X		X	X		
CATALOG	CA				X	X	X	X	X	X		X	X	X	
CICAT	HE				X		X	X	X	X		X		X	
KERNELC	92											X	X	X	
KERNELD	93											X	X		
MISCSEGC	95											X			
MEASSEG	96											X		X	
FILEIO	97											X	X	X	
DEBUGUTL	98														
INCLIO	A0														
INCLVMC	A1														
INCLPCB	A2											X			
INCLSLL	A3														
INCLST	A4														
INCLMEAS	A5											X			
INCLVMLD	A6														
INCLICS	A7											X			
INCLHARD	A8											X	X	X	
INCLMSG	A9														
RINSINCL	B0														

*= No longer exist after C.00.00

SYSTEM	LAST CHANGE NUMBER
B.00.00	0066
B.00.01	0134
B.00.02	0472
B.01.00	0789
B.01.01	1261
B.01.02	793-797,1283-1299,1400-1499
C.00.00	2056
C.00.01	2097
C.00.02	2727

NOTE: Each change made to MPE is now identified by a unique change number in columns 64/72 (eg <<00120>>). This matrix provides the range of the change numbers used to build each version of MPE.

II. SUPPORTED UTILITIES

A. UTILITIES MODIFIED

UTILITY	LEVEL
ASOCTABL	
DISKED2	00.00
*DPAN4	00.02
FREE2	00.00
IOCDPN0	00.00
LISTDIR2	00.00
LISTEQ2	00.00
LISTLOG2	00.00
PATCH	00.00
MEMLOGAN	00.00
MEMTIMER	00.00
SLPATCH	00.00
SPOOK	00.01
RECOVER2	00.00

* INDICATES UTILITY UPDATED/CHANGED BY THIS M.I.T.

*** END OF STATUS ***

Series 30/33/44 Software Update

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES 30/33
****

CONTENTS OF INSTALLATION TAPE DATE CODE '2131'

PRODUCTS WITH ASTERISKS ARE THE PRODUCT(S) UPDATED/CHANGED BY THIS M.I.T..

****

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE
*MPE	32033C	00.02	2131
SEGMENTER	32050A	01.03	2052
SPL	32100A	08.01	2052
BASIC	32101B	00.14	2052
FORTRAN	32102B	01.05	2052
BASIC COMPILER	32103B	00.14	2052
RPG	32104A	04.07	2052
BUILDINT	32150A	03.01	1623
*DS/3000	32190A	03.02	2131
*MRJE	32192A	01.03	2131
MTS	32193A	03.05	2052
DISCCOPY	32199A	00.01	2052
EDITOR	32201A	07.09	2052
SCIENTIFIC LIBRARY	32205B	00.04	1906
DEL/3000	32206A	01.10	2011
KSAM/3000	32208A	03.03	2101
*V/3000	32209B	02.00	2131
COMPILER LIBRARY	32211D	01.02	2052
FCOPY	32212A	03.13	2052
COBOL	32213C	02.07	2052
*SORT/MERGE	32214C	02.06	2131
*IMAGE	32215B	03.01	2131
QUERY	32216A	04.03	2011
IML	32229A	00.01	2052

CONTENTS OF INSTALLATION TAPE DATE CODE '2131'(Continued)

PRODUCTS WITH ASTERISKS ARE THE PRODUCT(S) UPDATED/CHANGED BY THIS M.I.T..

****

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE
*DIAGNOSTICS	32231A	2132	2131
COBOL LIBRARY	32232A	00.03	2052
COBOLII	32233A	00.03	2052
*OPT/3000	32238A	00.02	2131
DATACAPTURE PROCEDURES	32243A	01.00	2028
*DSG	32250A	00.01	2131
*DIAGNOSTICS	32340A	2132	2131
*TDP/3000	36578A	01.00	2131
*IFS/3000	36580A	00.01	2131
*IDS/3000	36581A	00.01	2131
PROG CONTROLLER 30300B/30361B-BCS	30361B	00.00	1621
PROG CONTROLLER 30301B/30361B-1-RTE	30361B-1	00.02	1701
RJE 2780/3780	30130E	01.00	2052

IFS/3000 and IDS/3000 were added on C.00.02

A. MODULES MODIFIED C.00.02

	MODULE		CHANGE HISTORY												
	NAME	NO	A.01.XX			B.00.XX			B.01.XX				C.00.XX			
			1	2	MR	0	1	2	0	1	2	3	0	1	2	3
	INITIAL	0	X		X	X	X	X	X	X	X		X		X	
	SYSDUMP	1	X		X	X	X	X	X	X	X		X		X	
	SEGPROC	2								X			X			
	SEGDVR	3								X			X			
*	DISPATCH	4				X			X	X						
	LOAD	5				X	X	X	X	X			X			
	UCOP	7				X	X	X		X			X		X	
	DEVREC	8				X	X	X	X	X			X		X	
	PROGEN	9	X		X	X	X	X	X	X	X		X		X	
	ININ	10	X	X		X		X	X	X	X		X	X	X	
	MEMLOGP	11				X			X		X				X	
	LOG	12		X		X				X	X		X			
	IOPTRDO	13				X					X					
	IOPTPNO	14		X		X					X		X			
	IOPLOTO	15				X					X				X	
	IOMDISCO	16									X					

(Continued)

	MODULE		CHANGE HISTORY												
	NAME	NO	A.01.XX			B.00.XX			B.01.XX				C.00.XX			
			1	2	MR	0	1	2	0	1	2	3	0	1	2	3
	IOFDISC0	17									X					
	IOTAPE0	18	X	X		X		X	X	X	X				X	
	IOLPRT0	19				X		X	X	X	X					
	IOCDRD0	20				X		X	X	X	X					
	HIOLPRT2	21														
	IOTERM0	22	X	X		X		X	X	X	X		X			
	IOPRPN0	24		X		X		X	X		X					
	HIOPPRT0	23													X	
	IOREM0	25									X					
	HIOTERM0	26									X		X	X		
	IOMDISC1	27		X		X				X	X		X			
	PFAIL	30						X					X	X		
	PVPROC	31				X	X			X	X		X			
	VINIT	32				X	X	X		X	X		X			
	SDFLOAD	33													X	
	SDFGEN	34									X					
	HIOTAPE0	35									X		X		X	
	HIOLPRT0	36									X		X			
	HIOMDSC1	37									X		X		X	
	HIOLPRT1	38									X					
	HIOFLOP0	39									X		X		X	
	MAKECAT	40				X		X	X	X	X		X			
	HIOTAPE1	41													X	
	FILEACC	50	X	X	X	X	X	X	X	X	X		X		X	
	COMM'INT	51	X	X	X	X	X	X	X	X	X		X		X	
	STORE/RESTORE	52	X			X		X	X	X			X		X	
	DIRC	53				X	X	X	X	X	X					
	ALLOCATE	54	X	X		X	X	X	X	X	X		X		X	
*	SOFTRES	56	X	X		X	X	X	X	X	X					
	HARDRES	55								X	X		X	X	X	
*	MMDISKR	57	X	X		X	X	X	X	X	X					
	ABORTDUMP	58		X		X		X		X			X			
	MESSAGE	59				X	X		X	X	X		X			
	PROCSEG	60	X	X		X		X	X	X	X		X			
	NRIO	62	X	X		X		X	X	X	X		X		X	
	PCREATE	63		X		X		X		X	X		X			
	MORGUE	64	X	X		X		X	X	X	X		X	X	X	
	BIPC	65											X			
	IPC	66											X			
*	DATASEG	67	X	X		X	X	X	X	X						
	CHECKER	69				X	X	X		X			X			
	UTILITY	70	X	X		X			X	X	X		X			
	SEGUTIL	71								X			X			
	LOADER1	72		X		X	X	X	X	X	X		X			
	RINS	73					X	X	X	X			X			

(Continued)

MODULE		CHANGE HISTORY													
NAME	NO	A.01.XX			B.00.XX			B.01.XX				C.00.XX			
		1	2	MR	0	1	2	0	1	2	3	0	1	2	3
JOBTABLE	74	X	X		X	X	X	X	X					X	
DEBUG	75				X		X	X	X	X		X			
NURSERY	76				X	X	X	X	X	X					
FIRMWARESIM	78				X							X			
SPOOLING	79	X	X		X		X	X	X	X		X	X	X	
SPOOLCOMS	80	X	X		X		X	X	X	X		X		X	
PVSYS	81				X	X	X		X	X		X			
UDC	82				X		X	X	X	X		X			
USER	83				X				X	X		X			
HELPUSER	84				X				X	X		X			
OPCOMMND	85							X	X	X		X	X	X	
LABSEG	86				X	X	X	X	X					X	
SDISC	87				X	X	X	X	X			X			
MEASIO	88							X	X	X		X			
LOGSEG0	90							X	X	X		X	X		
LOGSEG1	91							X	X	X		X	X		
CATALOG	CA				X	X	X	X	X	X		X	X	X	
CICAT	HE				X		X	X	X	X		X		X	
KERNELC	92											X	X	X	
KERNELD	93											X	X		
MISCSEGC	95											X			
MEASSEG	96											X		X	
FILEIO	97											X	X	X	
DEBUGUTL	98														
INCLIO	A0														
INCLVMC	A1														
INCLPCB	A2											X			
INCLSLL	A3														
INCLST	A4														
INCLMEAS	A5											X			
INCLVMLD	A6														
INCLICS	A7											X			
INCLHARD	A8											X	X	X	
INCLMSG	A9														
RINSINCL	B0														

*= No longer exist after C.00.00
**= New module added on C.00.02

SYSTEM	LAST CHANGE NUMBER
B.00.00	0066
B.00.01	0134
B.00.02	0472
B.01.00	0789
B.01.01	1261
B.01.02	793-797,1283-1299,1400-1499
C.00.00	2056
C.00.01	2097
C.00.02	2727

NOTE: Each change made to MPE is now identified by a unique change number in columns 64/72 (eg <<00120>>). This matrix provides the range of the change numbers used to build each version of MPE.

II. SUPPORTED UTILITIES

A. UTILITIES MODIFIED

UTILITY	LEVEL
ASOCTABL	
DISKED2	00.00
*DPAN4	00.02
FREE2	00.00
IOCDPNO	00.00
LISTDIR2	00.00
LISTEQ2	00.00
LISTLOG2	00.00
PATCH	00.00
MEMLOGAN	00.00
MEMTIMER	00.00
SLPATCH	00.00
SPOOK	00.01
RECOVER2	00.00

* INDICATES UTILITY UPDATED/CHANGED BY THIS M.I.T.

*** END OF STATUS ***

Editor's Note: The following article is a reprint of an earlier press release.

Computer Data Better Presented at Lower Cost by New Laser Printing System

A new, computer-controlled laser printing system from Hewlett-Packard fills an entirely new niche in the management of on-line distributed data-base applications. Although it serves office needs, its place is in the chain of an organizations' distributed computer system. The HP 2680 Laser Printing System produces computer output that is tailored to the needs of individual users. Business forms are electronically generated and automatically merged with data during the printing operation. These forms are quickly created, using an HP graphics terminal and new interactive HP software that makes it easy for non-specialists to design characters, logos, signatures and formats. Information may be from the data base of the associated HP 3000 Business Computer, or from any other HP 3000 networked to it. Electronic forms eliminate costly preprinted forms. Computer output no longer need be photoreduced on copiers for distribution, since everyone gets originals on manageable, people-size paper. Many different items can be produced in a steady stream, each in as many or as few copies as desired. The entire distribution cycle is simplified, so decision makers receive on-time information directly from the computer, in formats tailored to their needs.

Vast Variety of Effects for Better Information Presentation

The HP 2680A Intelligent Page Printer, under the control of an HP 3000 computer, uses a laser-based, electrophotographic process to print 45 pages per minute of z-fold paper, either U.S. 8 1/2 by 11 or European A4 size. Images may be placed anywhere on the paper. Character sets and electronic forms (up to 32 of each) may be used to organize page formats and highlight key information. The printer can print fonts as small as 22 characters per inch, to place more information on each page, saving paper and space. Large, bold characters can be used, to provide eye-catching headings and labels.

Information can be rotated 90 degrees on the paper, printing across the 8 1/2-inch dimension, so z-fold output can be read like a book, without bursting it apart. It is also possible to reduce print size so that four normal-size pages can be condensed to a single printed page. Clear, clean bar-codes can be printed, too.

A First: Software Implements CRT Composing

The Interactive Design System (IDS) software defines and modifies character sets, logos, and special symbols. Characters up to 1.38 inches in size are created by placing dots on a grid that appears on the associated HP graphics terminal. An HP 9874A Digitizer can be used to simplify the entry of complex symbols. With IDS, forms ranging from simple memos to complex invoices can be created. With the Interactive Formatting System software (IFS), multiple parts of an electronic form can be merged to replace expensive multi-part forms. It is thus that timely information can be tailored to the specific needs of each user.

Computer Compatibility

The HP 2680 Laser Printing System operates with HP 3000 Business Computer Systems, Series 30, 33, or 44, and Series III with the newly introduced HP-IB Interface Module. Because the printer incorporates a great deal of computer intelligence and memory of its own, it adds little to the overhead burden on the computer. Even the smallest HP 3000, it is estimated, could handle an HP 2680 System with 20 percent of its resources or less. Overhead typically involved in spooling out data to line printers has been shifted to the operating system of the HP 2680, itself.

Thus the controlling HP 3000 Computer also can be expected to perform all of the data-processing functions normally expected of it.

Minimized Total Cost of Ownership

The HP 2680A Intelligent Page Printer was designed to produce a mean of 100,000 copies between failures, far above the usual figure for other printers using the same principle of operation. A microcomputer manages all mechanical operations, controlling interlocks, paper movement, forms registrations and stacker adjustments. The same microcomputer monitors print quality continuously, and makes appropriate adjustments of power supplies and toner mixture to keep the copies clean. A message display panel alerts the operator when maintenance is due. Built-in self-diagnostics speed normal, periodic servicing.

Output is produced on plain bond continuous-fanfold paper. Weights from 18-pound to 24-pound bond are usable. Paper lengths may be from 3 to 17 inches (in 1/2-inch increments), if desired, and widths may vary from 6.5 to 12.7 inches. The printing system senses the paper width, and limits printing to that surface area of the photoconductive drum and fusing apparatus, to improve reliability and reduce energy consumption. Infrared fusing of the toner uses no mechanical parts, and involves no direct contact with the paper.

Total cost per copy, including amortization, supplies and maintenance, is expected to range from about six cents, at 50,000 copies per month, to somewhat less than three cents at 400,000 copies per month.

The 2680A Laser Printing System from the Programmer's Perspective

by Jim Langley, Boise Division

Editor's Note: The following article is taken from a paper on the Hewlett-Packard 2680A Laser Printing System which was prepared for the HP 3000 User's Group Meeting on April 29, 1981 in Orlando, Florida.

Abstract

This paper describes the HP 2680A Laser Printing System from the perspective of the HP 3000 programmer. The printer hardware is first described, then its features are explained. Concepts of downloadable character sets, electronic forms and logical pages are discussed. The implementation and use of these printer features via the system software is also covered. The impact of the laser printer in a distributed computing environment is briefly explored.

Overview

The HP 2680A is Hewlett-Packard's first page printer. It is based on an electrophotographic process which was licensed from Canon, a Japanese firm. The printer was designed and is manufactured by the Boise Division in Boise, Idaho.

Several key objectives were established at the start of the program. Reliability, flexibility, features matched to 3000 user needs, simple powerfail and paper jam recovery, very low CPU overhead, and the ability to access the printer and its features from existing programs without modification became the primary objectives of the development effort.

From the beginning, the printer was designed as an extension of MPE, not an added-on peripheral. This tight coupling yielded a fully integrated printing system that is fully supported by the MPE file system and spooler. In addition, a powerful subsystem exists which allows complete character set and forms design. Flexible page formatting and a full complement of intrinsics provide access to all printer features.

By fully integrating the printer into the 3000, simple and reliable power fail and paper jam recovery is realized. All these benefits were achieved while the CPU overhead to drive the printer was reduced an order of magnitude from that required to drive conventional printers at comparable data rates.

Hardware

The 2680A is approximately 5.5 feet long, 2.5 feet deep, and 4 feet high. It weighs about 875 pounds. Power requirements are 4500 watts when printing. Throughput is 45 8.5 by 11 inch pages per minute. The equivalent lines per minute speed is 2900 lpm ranging up to 12,000 lpm in reduction mode.

The paper path is short and readily accessible to the operator. It features a powered paper stacker. The fusing system is radiant, eliminating any pressure or high temperature rollers. Nothing contacts the upper side of the paper once the image is transferred from the drum to the paper, contributing to excellent reliability. The web is pulled by a programmable torque motor on the output tractors, and paper motion is gated by stepper motor driven input tractors. A solenoid powered retraction mechanism pulls the paper away from the drum when the seam on the drum comes around. The input paper platform acts as a splice table; a vacuum is used to hold the paper onto the table when splicing a new box of paper onto the end of the previous box. The paper path can accomodate various widths up to 12.5 inches and lengths up to 17 inches. A width sensor on the input tractors allows the printer to energize only the correct width in the preheater section of the fuser. Paper which is heated produces odors, which are trapped and oxidized in replacable filter cartridges.

The image forming process is electrophotographic. The heart of the system is a photoconductive drum about 19 inches in circumference and 14 inches long. The drum is coated with cadmium sulfide and wrapped in mylar for protection. The drum is uniformly erased and then charged to several hundred volts at the first station. The laser is then scanned across the drum perpendicular to the direction of rotation. The beam is modulated to give 180 dots per inch resolution. There are 2048 dots in one scan line, giving a printable area 11.38 inches across. The drum rotation allows the sweeping laser beam to cover an area 17 inches long. The circular dot is about .008 inches in diameter on a grid .0055 inches square. When the laser beam hits the drum the voltage is depleted. Next, the drum rotates past a cloud of fine flour-like black plastic. The plastic toner is attracted to areas of no voltage by electrostatic forces. The pattern traced by the scanning laser beam is now visible as a sharp black image on the drum. Finally, the paper and the drum are brought together for about 1 inch of tangential contact. The paper is correctly charged to firmly attract the toner off the drum. The small amount of residual toner not deposited on the paper is then scraped off of the drum by a urethane wiper blade and collected by a vacuum system. As the drum turns, these processes are executed simultaneously at different stations around the drum.

In order to achieve high print quality over a wide range of ambient conditions, the HP 2680A has two closed loop control systems. The electrostatic control system monitors the potentials on the drum just after the laser station. The voltage is mea-

sured both where the laser exposed the drum, and where it did not. The microprocessor taking the measurements then controls several programmable power supplies to maintain the correct drum potentials. The readings and adjustments are made once per drum rotation. The electrostatic closed loop compensates for variations between replacement drums, drum degradation over time, humidity, temperature and altitude variations, and toner mixture fatigue.

A second closed loop system monitors the developed image on the drum to control print density on the page. By varying the amount of toner in the developer assembly which brushes the toner mixture accross the drum, the amount of toner on the drum, and hence the final print darkness on the paper, can be controlled.

The mechanical features of the printer were designed to be simple and reliable, and the operator functions are easy to learn and execute. A vacuum system in the printer contributes to cleanliness. It is used to recover toner wiped off the drum. It also is used for the splice table and to maintain good contact between the preheater pad in the fuser and the paper. The operator loads a fresh kilogram of toner into the machine about once every eight hours of printing. Unused toner is collected with the vacuum system, trapped centrifigally, and deposited in a disposable bottle which is replaced every couple of days. A new box of paper is loaded every hour. The new box can be conveniently spliced onto the end of the previous box, or the new box can be easily loaded with the THREAD button.

There are two microprocessors in the HP 2680A. One is a 16 bit HP proprietary SOS device which controls all machine functions such as the operator keyboard and alphanumeric display, the paper path, the closed loop systems, and internal diagnostics. The second processor is a high speed, bi-polar bit slice processor which communicates with the 3000 and performs all processing on the data stream, and ultimately modulating the laser beam to form the correct images at the proper place on the drum. This processor uses 256K bytes of RAM, with a second 256K available as an option. Approximately 40K bytes of this memory is used for tables and buffers. The remaining memory is partioned dynamically during each job for character sets, forms, and page buffering.

Extensive internal diagnostics constantly monitor the state of the machine, alerting the operator if a service engineer should be called. When arriving on site, the service engineer can use additional internally contained diagnostics to troubleshoot any problems. A very complete self test program is available which prints many important parameters on the machine itself. Data such as serial number, drum rotations since last PM, firmware datecodes, and all operating values are labeled and printed. The printer contains a limited amount of nonvolatile memory.

Programming Features

Page printers, even with their inherent benefits of high thruput, low noise, and exceptional print quality are rarely viable as simple print and space devices because of their higher cost. However, the HP 2680A is a cost effective replacement for many line printers. This is because of the flexibility and features of the printer. Electronic forms allow the inventory of costly spceialty forms to be eliminated. Long lead times and form modification costs are reduced to a few hours on a terminal. Definable character sets allow the printer to be used in a wide variety of industries and applications where conventional printers are useless. In addition, the print quality and crispness in conjunction with the 8.5 by 11 inch paper size means HP 2680A output never needs to be copied or reduced before general distribution.

The HP 2680A implements a concept called logical pages. A physical page is a sheet of paper bounded by perforations. A physical page can be divided into up to 32 rectangular areas called logical pages. Logical pages can overlap. A programmatic command to page eject moves the print to the next logical page. If all logical pages have been used, the printer goes to the first logical page on the next sheet of paper. Each logical page has several attributes such as an associated vertical format control (VFC) table, a default line spacing, and one of four orientations. In addition, each logical page can have up to two forms associated with it. When the logical page is printed, the forms are automatically overlaid by the printer. Several logical pages can share the same form and VFC; the printer will automatically relocate it to the correct origin for each logical page. Logical pages are a powerful concept which particularly supports existing programs. By defining the logical page format, an existing job can have its output reduced two to one, or four to one, or rotated without even recompiling the job. Additionally, a job which currently uses preprinted forms can be switched to run on the laser printer without modification. The existing form is converted to electronic format and then the corresponding logical page is defined to use the form. The job is then printed on the laser printer and the data is merged with the form and printed.

The electronic forms capability is designed for maximum flexibility. Each form can contain horizontal and vertical lines of varying thicknesses, text written with any number of fonts in any of the four basic directions, plus areas or boxes of variable shading. Form elements can be positioned anywhere and are not restricted to certain character positions on the page as a "draw set" is. The printer can support 32 different forms simultaneously. Each logical page can use up to 2 forms as long as the total does not exceed 30. Additionally, each physical page can be overlaid with up to 2 forms. Enough memory and processing

power exists to create a form which is a dot per bit image of an 8.5 by 11 inch sheet of paper. Forms are easily created for the printer using an interactive program called IDSFORM.

The HP 2680A printer accepts user defined character sets. Each character set contains from 1 to 128 characters. Each character has an associated cell of a specified size which contains any dot per bit representation desired. The spacing between characters and between lines can be set to any value. A character set can print in any of the four directions. Proportional character sets are supported. In this case, each character has a parameter describing how far to move over after printing each character. The printer also allows the cells to be printed in any relationship to the current "pen" position. This allows centered symbols, or common base lines so different character sets can be mixed properly on a single line. When using more than one character set, a primary and secondary set are defined and then selected with either shift in, shift out control codes, or by setting the eighth bit of the ASCII code. HP supplies a large number of character sets of various fonts and sizes. In addition, character sets and logos can be created interactively by terminal users via IDSCHAR.

Thirty-two user definable VFC's are supported by the printer simultaneously. They are easily created with the IFS2680 program.

One additional feature was implemented to allow easy emulation of multi-part forms. When activated, each physical page of data will be repeated up to eight times by the printer. As each copy of the page is printed, the printer will automatically overlay any two forms on the page. In this manner, the same data can be repeated up to eight times, but each copy can be individually addressed to shipping, purchasing, order processing, etc.

These basic data structures provide a wide range of user features. When combined with the ability to place cells anywhere on the page and overlap at will, plus the processing power to handle over 20,000 characters on an 8.5 by 11 inch sheet, a truly unique printer results. The maximum number of cells on any raster scan is 255. As the cells get larger, fewer can be printed simultaneously. Character set switching, forms overlay and other features all occur at speed.

The printer's memory is allocated by a memory manager on a job by job basis. Approximaetly 40K bytes are used by the printer. The remaining memory is allocated to character sets, forms, VFC's, and page buffering. As much memory as required is allocated to the user's character sets, forms, and VFC's. All remaining memory is used to buffer pages in an intermediate linked list structure. More page buffering insures that pages are printed at speed. Insufficient page buffering causes a lower thruput rate. The programmer can add or delete character sets, forms, and VFC's during the job.

Environment Files

All character sets, forms, VFC's, and the logical page table and the multicopy forms table are placed in an environment file by a terminal user running IFS2680. This file is then sent to the printer at the start of a job automatically. This allows the output of a job to change appearance by changing the environment file or portions of the environment file. For example, if the character set in an environment file is changed from elite to pica, the next job to use the file will have output printed in pica. By simply changing the logical page description and substituting a smaller character set, a job can be made to print in a two-to-one or four-to-one reduction mode. HP supplies several standard environment files to cover portrait mode pica and elite, landscape 132 column printer emulation, two-to-one and four-to-one reduction. The user can easily create additional environment files.

For new application programs, the full power of the printer is available through HP supplied intrinsics. The intrinsics allow features such as writing a string to a named field on an electronic form. The form can be redesigned and rearranged without modification of any programs using the form. The data will automatically be placed in the correct field wherever it is on the page. Intrinsics also allow the pen to be moved, new primary and secondary character sets to be selected, any logical page to be turned on or off, and other similar features.

System Software

Extensive system application software allows creation of character sets, forms, and VFC's, as well as the definition of logical pages and multicopy forms tables.

IDSCHAR provides menu driven interactive creation of character sets on graphic terminals. The program can emulate various shaped dots and grid spacings. The laser printer has round dots about 8 mils in diameter on a 5.5 mil grid. IDSCHAR also supports a digitizer to allow easy input of character or logo outlines. The outline can then be scaled and presented superimposed on the cells grid for easy filling in. IDSCHAR supports lines, arcs, rectangular area fills, plus scanning and rotation. Special logo files are supported for use on forms. An experienced graphics designer can create a complex logo in one to four hours. Generating a high quality character set takes about 40 hours.

IDSFORM provides menu driven interactive forms creation of forms on graphics terminals. It supports horizontal and vertical lines of three different thicknesses. Boxes can be shaded from clear to black. IDSFORM supports subforms which can be defined and then easily moved around, both on the page, and between different forms. Windows describe boxes consisting of headers and data

allowing the form to be changed around without modifying the program. The 1040 tax form was perfectly emulated in 14 hours by an experienced user of IDSFORM.

IFS2680 is the formatting program which bundles up different character sets, forms, VFC's, and a logical page table into an environment file. IFS2680 also is the program which constructs VFC's and the logical page table for the user. Overall job parameters, such as the number of copies of each page desired and the multicopy forms table, are specified via IFS2680. HP supplied standard environments are available from IFS2680 either as they stand, or as a base to begin creating a unique environment for a special job.

A contributed program called TR2680, which interprets commands imbedded in ASCII files, is available. Text editors can be used to prepare memos and reports with the imbedded commands to utilize HP 2680A features such as multiple character sets, forms overlay, pen moves, etc.

Once an environment file is creted, it is specified with a new option in the file equation, ":FILE PRINT;DEV=PP;ENV=FOURTO1". The environment file is automatically placed in the spool file before the data. This allows existing programs to use all of the features accessible via environment files without modification.

Power fail and jam recovery are very simple and reliable. Non-volatile memory exists in the printer. When power resumes the 3000 retransmits data and resumes printing at the correct point in the job. The only operator intervention required is to insure top of form is correctly aligned, and push RUN. Paper jams are similar. If no paper was damaged, the job can be resumed without system intervention. If the operator wishes to back up several pages, the spooler is suspended, the jam cleared, and the command ":RESUMESPOOL LDEV#;BACK 5 PAGES" is used. This allows backing up or skipping forward an arbitrary number of pages.

Another unique concept introduced with the laser printer is the error trailer. When a program executes an illegal function such as selecting a missing character set, moving the pen off of the logical page, or trying to print a character off of the logical page, the printer relays this information to the 3000. This information is then printed out at the end of the job before the trailer is printed. The error trailer describes the error in English, along with the record number and actual page number where the error occurred.

Distributed Printing

MRJE has been modified to support HP 2680 environment files. If the device class is PP for page printer, and the forms field is not empty, then the forms identifier is used to locate an environment file.

RJE has an option which allows the translator procedure to process each record when received by the 3000. This allows complete access to the printer's features from a mainframe.

One internal test site is running a Series 30 to front end the laser printer. They are printing over 130,000 pages per month. One half of the output is generated by an Amdahl 470, and sent at 9600 baud via MRJE.

At 2900 lpm, the printer taxes the performance of most data communication systems. System configuration, CPU overhead, and data format determine the printer utilization. The range can be from 10% to 100%. We are currently quantifying printer performance in these areas, and welcome user inputs and insights.

Summary

The HP 2680A laser printing system provides a cost effective solution to many computer output problems for HP 3000 users. The reliability and servicability contribute to its low cost of less than four cents per page at 200K pages per month. The unmatched features provide capabilities unique in the industry. The complete software application package allows immediate turnkey solutions with no programming. The impact of the laser printer in the distributed network is significant and allows non HP systems to utilize the printer, as well as enhancing distributed HP systems.

Introducing IDS/3000and IFS/3000 Design and Formatting Software for the 2680 Laser Printing System

by LaVerne Burnham, Information Networks Division

The 2680 Laser Printing System combines the HP 2680A Intelligent Page Printer with powerful output design and formatting software. As shown in Figure 1, the 2680 Laser Printing System consists of:

- IDS/3000
  - IDS/CHAR
  - IDS/FORM
- IFS/3000
  - IFS/2680
  - Programmatic Interface
  - Supplied Character Fonts
  - Supplied Environments
- 2680A Intelligent Page Printer

The IDS/3000 and IFS/3000 subsystems enable you to use the 2680A Page Printer to print a specific document such as a memo or report. With IDS/3000, the Interactive Design System, you can design your own characters, logos, and forms to meet your specific needs. With IFS/3000, the Interacive Formatting System, you can format the layout of your document and select the characters, logos, and forms needed to print the document. In addition, Hewlett-Packard supplies a number of the most commonly used character fonts and formats (called environments).

IFS/2680 and the Programmatic Interface are focused specifically at the 2680A Page Printer, enabling you to take full advantage of the capabilities of the 2680A. With IFS/2680, you combine character fonts or logos and forms designed with IDS/3000 together with instructions for the layout details of your document. Additional control of the 2680A is provided by the Programmatic Interface, which consists of high-level procedures callable from COBOL, BASIC, FORTRAN, and SPL.

The 2680A Laser Printing System provides you with a high degree of flexibility in printing your documents. You can use multiple character fonts in various sizes. For example, using a character font smaller than the standard line printer font enables more information to be printed on a page. Also, the characters can be rotated so that the information can be printed on notebook size paper (8-1/2 by 11 inches).

You can use the character fonts and logos designed with IDS/CHAR in your form. With the electronic forms capability, you print the form as well as the information in the form simultaneously. You can also combine multiple forms to simulate a preprinted multicopy form.

You can group the information in your document into modular units. You can then vary the rotation as well as the location of these units on the printed page. This simplifies the formatting of documents to meet your specific needs.

With the Programmatic Interface, a user-written application program can direct the 2680A in positioning data and in selecting character fonts and forms. For example, the Programmatic Interface provides symbolic reference to forms for the purpose of printing data in the form, rather than reference by location on the page.

The following features of the subsystems make character and forms design and document formatting easier:

- Interactive systems for defining your specifications at a terminal using fill-in-the-blank menus and function keys.

- System supplied default values for the most commonly used specifications. If desired, you can change these defaults.

- Application-oriented design and formatting process. You use the terminal to actually draw characters and forms on the terminal screen.

- Easy modification of existing characters, forms, and document formats.

Characters and forms can be designed on 2647A and 2648A terminals. Templates and terminal key reference cards are provided as a quick guide to the functions performed by the terminal keys. You can also use the 9874A Digitizer and the 9111A Graphics Tablet as aids in designing characters. In addition to the terminals listed above, you can run IFS/2680 on the 2640B, 2645A, and the 2641A.

A 4-day customer training course is available and is designed for forms and character designers and for programmers. The course provides a basic orientation to the Laser Printing System as well as definition and potential application of terms that are specific to the system. The major emphasis of the course is a series of modular hands-on exercises. In these self-paced exercises, the student designs a character font or logo and a form, defines the layout details, and then prints the resulting document on the 2680A.

The reference guides defining the design and formatting software are listed below. These guides are in a comfortable 9x9 size binder. The loose-leaf pages can easily be removed for changes and insertions of material.

The IDS/CHAR guide is primarily for graphics designers and others who have a basic understanding of typesetting terminology. Similarly the IDS/FORM guide is for those whose function is to design forms. The IDS/3000 guide may be used by many different users with or without programming experience; their common goal is to use the capabilities of the 2680A to print specific documents.

The following is a list of part numbers for the customer course, and terminal and digitizer overlays. Also included are part numbers for both the hardware and software reference manuals.

CUSTOMER TRAINING COURSE

2680 Laser Printing System Course.....P/N 22838A/X

OVERLAYS

IDS/3000 Templates for the HP2647A.....P/N 36581-60001
IDS/3000 Templates for the HP2648A.....P/N 36581-60002

REFERENCE MANUALS

IDS/CHAR Reference Guide.....P/N 36581-90001
IDS/FORM Reference Guide.....P/N 36581-90002
IFS/3000 Reference Guide.....P/N 36581-90003
2680A Operator's Handbook.....P/N 02682-90901
2680A Service Manual.....P/N 02682-90904

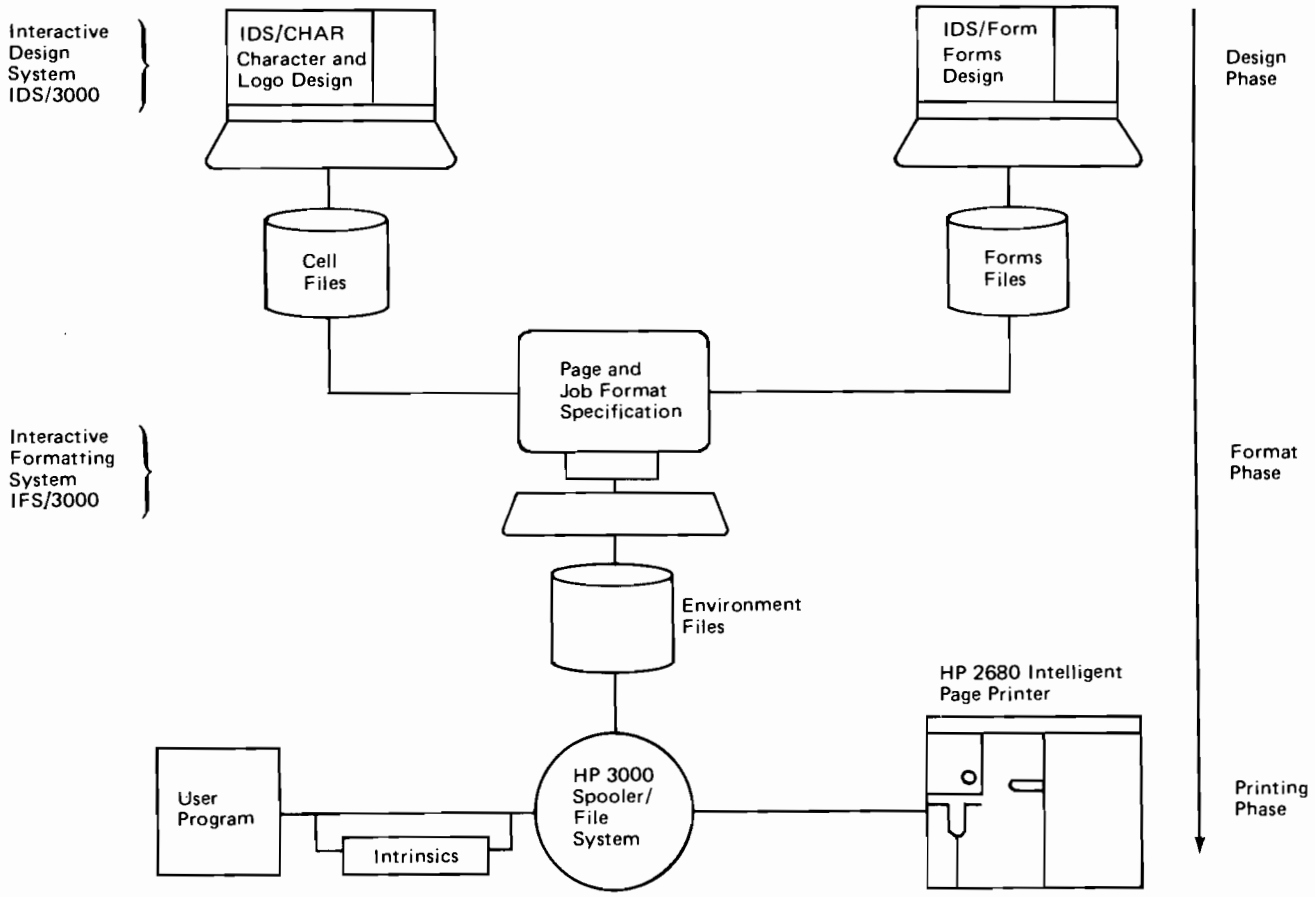


Figure 1. HP2680 Laser Printing System

Using IFS/3000 Intrinsics

by Carolyn Spitz, Computer Systems Division
and John Ramuta, Bay Area EDP

IFS3000 intrinsics are the set of procedures that control the HP2680A Laser Page Printer. Using these intrinsics, you can programmatically change forms, change print direction, use alternate character fonts, and many other operations. The HP2680A, together with the IFS3000 intrinsics, provides much more flexibility than normal line printers. In addition to traditional "print-and-space" applications, you can write to forms by referencing the various fields with their symbolic names.

To use the intrinsics, you need to create an environment file with IFS2680 that contains the forms, character sets, etc., to be referenced by the program. (When using environments, it may be necessary to "PREP" programs with larger MAXDATA than was previously needed.) The intrinsics are callable from COBOL, FORTRAN, BASIC and SPL. This article will describe six frequently used IFS3000 intrinsics with explanations of the parameters required. A simple programming example is provided for each language.

Some basic rules apply to all intrinsic calls. All parameters are passed by reference, and all parameters are required. Each intrinsic has a communication area as its first and perhaps only parameter. PINITIALIZE must be the first IFS3000 intrinsic an application program calls; it is only called once. It initializes the communication area that is used to pass information among the intrinsics and to the user. The "COMAREA" also contains information about character fonts, logical pages and data which allows symbolic access to fields within forms.

The communication area is a logical array (SPL conventions will be used when discussing data types) and is passed as the first parameter of PINITIALIZE. Its size will depend on the environment file but it should be at least as many words as were displayed on the IFS2680 main menu when the environment was compiled. The length is passed as an integer variable in the second parameter position. To accomodate for differences in the various programming languages, (e.g. how byte arrays are passed) PINITIALIZE requires as its third parameter an integer variable identifying the calling language. The variable should equal 0 if using COBOL/3000, 1 if using BASIC, 2 if using FORTRAN, and 3 if using SPL or COBOLII with the CALL INTRINSIC. The last parameter is a integer variable containing the system file number associated with the environment file that is being initialized.

The following programming examples show the data definitions that will be used in the discussions of the intrinsics as well as procedures for opening the files and calling PINITIALIZE. For simplicity of discussion, these examples will assume the following file statement was entered before execution:

```
:file OUTFILE;dev=DEV2680,env=ENVFILE
```

where OUTFILE is the formal file designator used in the programs, DEV2680 is the device class of the HP2680A, and ENVFILE is an environment file created using IFS2680. In the FORTRAN example, assume:

```
:file FTN07;dev=DEV2680;env=ENVFILE;cctl
```

COBOL

```
file-control.
    select   OUT-FILE     assign to  "OUTFILE".
data division.
file section.
01  OUT-FILE
    label records are omitted.
01  OUT-RECORD       PIC X(132).
working-storage section.
01  COMAREA.
    03  USERSTATUS  pic S9(4)   comp.
    03  ERRNUM      pic S9(4)   comp.
    03  SYSTEM-INFO pic S9(4)   comp   occurs 1998 times.
01  LANGUAGE        pic S9(4)   comp   value 0.
01  AREALENGTH      pic S9(4)   comp   value  2000.
01  FORMNAME        pic X(16).
01  FIELDNAME       pic X(16).
01  SUBFIELDNUM     pic S9(4)   comp   value -1.
01  DATABUFFER.
    03  filler      pic X(15)   value "This is line 1.".
    03  filler      pic S9(4)   comp   value 13.
    03  filler      pic X(15)   value "This is line 2!".
    03  filler      pic X       value  high-value.
    03  filler      pic X(100)  value "H-P".
01  DATALENGTH      pic S9(4)   comp   value 100.
01  ERRMSG          pic X(80).
01  ARRAYLEN        pic S9(4)   comp   value 80.
01  MSGLEN          pic S9(4)   comp.
procedure division.
A100-initialize.
     open output OUT-FILE.
     call "PINITIALIZE" using  COMAREA, AREALENGTH,
                               LANGUAGE, OUT-FILE.
```

BASIC

```
  10 REM ************************************************
  20 REM *  A   (AREALENGTH)
  30 REM *  C1  (COMAREA)
  40 REM *  D   (DATALENGTH)
  50 REM *  D$  (DATABUFFER)
  60 REM *  E1  (ARRAYLEN)
  70 REM *  E2  (MSGLEN)
  80 REM *  E$  (ERRMSG)
  90 REM *  F   (FNUM)
 100 REM *  F1$ (FIELDNAME)
```

```
110 REM *  F2$ (FORMNAME)
120 REM *  L   (LANGUAGE)
130 REM *  S   (SUBFIELDNUM)
140 REM ************************************************
150 INTEGER C1[2000],L,A,S,D,E1,E2,F
160 DIM D$[100],F1$[16],F2$[16],E$[80]
170 A=2000
180 L=1
190 REM ************************************************
200 REM * THE NEXT 3 STATEMENTS CAUSES
210 REM *  -OUTFILE TO BE OPENED AT START OF EXECUTION
220 REM *  -THE SYSTEM FILE NUMBER RETURNED
230 REM *       ASSIGNED TO AN INTEGER
240 REM *  -THE COMMUNICATION AREA TO BE INITIALIZED.
250 REM ************************************************
260 FILES OUTFILE
270 F=SFN(1)
280 CALL PINITIALIZE(C1[*],A,L,F)
```

FORTRAN

```
      $CONTROL FILE=7
      C The FILE=nn causes FTNnn to be placed in the FLUT
      C
            integer    COMAREA(2000),USERSTATUS,LANGUAGE,
           *           AREALENGTH,SUBFIELDNUM,DATALENGTH,
           *           FNUM, ERRNUM, ARRAYLEN, MSGLEN
      C
            equivalence (USERSTATUS,COMAREA(1)),
           *            (ERRNUM,COMAREA(2))
      C
            character*100 DATABUFFER
            character*16  FIELDNAME,FORMNAME
            character*80  ERRMSG
      C
            language=2
            arealength=2000
      C FNUM is an HP3000 compiler library procedure which allows
      C a FORTRAN program to get the system file number from the
      C FLUT.  FNUM will open the file if no file is open for the
      C specified Unit.  Then the COMAREA is initialized.
           call PINITIALIZE(COMAREA,AREALENGTH,LANGUAGE,FNUM(7))
```

SPL

```
     logical array COMAREA(0:1999),
                   OUTFILE(0:4),
                   FORMNAME(0:7),
                   FIELDNAME(0:7),
                   ERRMSG(0:39);

     byte array    FIELDNAME'(*)=FIELDNAME,
                   FORMNAME'(*)=FORMNAME,
                   ERRMSG'(*)=ERRMSG;

     define        USERSTATUS=COMAREA(0)#,
```

```
                    ERRNUM=COMAREA(1)#;

     equate         EOS=255, CR=13;

     integer        AREALENGTH:=2000;
     integer        LANGUAGE:=3;
     integer        SUBFIELDNUM:=-1;
     integer        DATALENGTH:=100;
     integer        ARRAYLEN:=80;
     integer        FOPT:=%407;
     integer        AOPT:=%101;
     integer        FNUM, MSGLEN;

     intrinsic      FOPEN, PINITIALIZE, PWRITEFIELD, PNEWPAGE,
                    PERRMSG, PNEWFORM, PNEWPHYSPAGE;

     begin
          move OUTFILE:="OUTFILE ";
          FNUM:=FOPEN(OUTFILE,FOPT,AOPT);
          PINITIALIZE(COMAREA,AREALENGTH,LANGUAGE,FNUM);
```

PWRITEFIELD

PWRITEFIELD allows you to write data to fields or subfields that were defined during forms design using IDS/FORM. Since you can write to fields using the symbolic field names, you do not have to be concerned with their positioning on the form. In fact, you can move the fields on the form without affecting the application programs that write to them.

One or more lines of data can be written into a single subfield using PWRITEFIELD. The array containing the data to be written to the field can be as large as you want, but each print line has a maximum of 250 bytes. Print lines are separated in the data array by a carriage return (ASCII 13, octal 15). A carriage return is not required in the last line of data unless that line of data is blanks. When data is printed in the subfield area it is centered verically and left justified. Therefore, you may wish to include blank lines or leading blanks within a line to position the data. Trailing blanks are stripped from each data line before it is printed. The length of the data array is passed in the PWRITEFIELD call. You may, however, terminate a data array before the end of the array is reached by using trailing blanks or an EOS (End of String: character 255, octal 377).

PWRITEFIELD takes 5 parameters. The first is the logical array specifying the communication area. The second is the byte array indicating the field name which is to be written to. This is the field name specified in IDS/FORM and may be up to 16 characters. If it is less than 16 characters it should be terminated by a blank or an EOS. (In BASIC a terminating character is not needed.) The third parameter is an integer variable containing the subfield number you want to write to. If there is only one subfield in the field you should use a -1. Otherwise the subfields are numbered beginning at one. The variable passed should

not be greater than the number of subfields in the field or an error will occur. The byte array containing the data to be written is passed as the fourth parameter. The fifth PWRITEFIELD parameter is an integer variable indicating the length (in bytes) of the data array you are writing to the field.

The following examples show how a program might write to a field that has only one subfield. Two lines are written from a large data array (100 bytes long) using an imbedded carriage return and EOS to terminate the array. The lines are printed as follows:

```
This is line 1.
This is line 2!
```

The lines are left justified and centered vertically in the field. Notice that "H-P" is not printed because an EOS character is used to terminate the data before the end of the array in all the examples. (See data definitions in PINITIALIZE examples.)

COBOL

```
     move "FIELD1 " to FIELDNAME.
     call "PWRITEFIELD" using COMAREA, FIELDNAME
                   SUBFIELDNUM, DATABUFFER, DATALENGTH.
```

BASIC

```
      700 F1$="FIELD1"
      720 S=-1
      730 D=100
      730 D$="This is line 1. This is line 2! H-P"
      740 D$[16,16]=CHR$(13)
      750 D$[32,32]=CHR$(255)
      760 CALL PWRITEFIELD(C1[*],F1$,S,D$,D)
```

FORTRAN

```
      FIELDNAME="FIELD1 "
      SUBFIELDNUM=-1
      DATABUFFER="This is line 1. This is line 2! H-P"
      DATABUFFER[16:1]=%15C
      DATABUFFER[32:1]=%377C
      DATALENGTH=100
      call PWRITEFIELD(COMAREA,FIELDNAME,SUBFIELDNUM,
     *                      DATABUFFER,DATALENGTH)
```

SPL

```
      move FIELDNAME:="FIELD1 ";
      move DATABUFFER:=("This is line 1.",CR,
                        "This is line 2!",EOS,"H-P");
      PWRITEFIELD(COMAREA,FIELDNAME',SUBFIELDNUM,
                        DATABUFFER',DATALENGTH);
```

PNEWFORM

PNEWFORM is used to make a form accessible to the calling program. If only one form exists for the current logical page then it is not necessary to call this intrinsic. If more than one

form, or a multicopy form was set up, the PNEWFORM intrinsic must be called to specify which form you wish to use.

The intrinsic requires 2 variables, which we will call "COMAREA" and "FORMNAME". The makeup of COMAREA has already been described in the PINITIALIZE intrinsic information.

FORMNAME is a byte array of up to 16 characters. If it is shorter than 16 characters it should be terminated by a blank or an EOS character. The exception to this is in BASIC, where no blank or EOS is required, since BASIC passes the length of the string to the intrinsic. The first character of the string must be a letter. There is no difference between upper and lower case.

For example, if you were to choose to write to a form called "INVOICE" the PNEWFORM call may look as follows:

COBOL

```
     move "INVOICE " to FORMNAME.
     call "PNEWFORM" using COMAREA, FORMNAME.
```

BASIC

```
     600 F$="INVOICE"
     610 CALL PNEWFORM(C1[*],F2$)
```

FORTRAN

```
     FORMNAME="INVOICE "
     call PNEWFORM(COMAREA,FORMNAME')
```

SPL

```
     move FORMNAME := ("INVOICE ");
     PNEWFORM(COMAREA,FORMNAME');
```

PNEWPAGE
PNEWPAGE is used to move to the next logical page. If there are no more logical pages, a physical page eject occurs and the first logical page is again enabled. A logical array specifying the communication area is the only parameter required for PNEWPAGE.

COBOL
```
     call "PNEWPAGE" using COMAREA.
```

FORTRAN
```
     call PNEWPAGE(COMAREA)
```

BASIC
```
     440 CALL PNEWPAGE(C1[*])
```

SPL
```
     PNEWPAGE(COMAREA);
```

PNEWPHYSPAGE
Using PNEWPHYSPAGE is similar to using PNEWPAGE. The difference is that PNEWPHYSPAGE always causes a physical page eject, and resets everything to the first active logical page and form. This is useful when you want to skip to a new physical page immediately, even if all logical pages haven't been used. PNEWPHYSPAGE is called in the same manner as PNEWPAGE.

PERRMSG
The status of each intrinsic call is returned in the first word of the communication area. Successful calls will return a value of 0. Unsuccessful calls will return 1 if a warning was issued and 2 if an error occurred. After each intrinsic call you should check if the intrinsic completed successfully. If it didn't, use the PERRMSG intrinsic to get the associated error message. Since PERRMSG uses the IFS2680 message catalog (PMSGCAT.PUB.SYS), this should be resident on the system. When PERRMSG completes, it will reset the first five words of the communication area to zero. If PERRMSG fails, however, the words are not reset and the error number can be obtained from the second word in the communication area.

PERRMSG takes 4 parameters. The first is a logical array specifying the communication area. The second is an array where the message is returned. In the third parameter you pass the length of the error message array as an integer variable. The array should be at least 80 bytes long to avoid truncation. The fourth parameter is the integer variable where the actual length of the message is returned.

The following examples show how PERRMSG might be used in a program.

COBOL
```
     if USERSTATUS not = 0
```

```
      call "PERRMSG" using COMAREA, ERRMSG, ARRAYLEN, MSGLEN.
```

```
BASIC
     550 if C1[1]<>0 then call PERRMSG(C1[*],E$,E1,E2)
     560 REM    The last two parameters are required but
     570 REM    not used since BASIC determines the lengths
```

```
FORTRAN
     if (USERSTATUS.NE.0)
       call PERRMSG(COMAREA,ERRMSG,ARRAYLEN,MSGLEN)
```

```
SPL
     if USERSTATUS <> 0 then
       PERRMSG (COMAREA,ERRMSG',ARRAYLEN,MSGLEN);
```

These were just six of the seventeen IFS3000 intrinsics now available to application programs. Others allow you to select character fonts (PUSEFONT), control the location of printing (PMOVEPENABS and PMOVEPENREL), and other operations that make IFS3000 intrinsics a powerful tool. Although it takes a little initial effort to learn the intrinsics, you will find they are simple to use and the results very rewarding.

Using Environment Files

by Stephanie Littell, Computer Systems Division

With the addition of the 2680A Page Printer to your installation, there is a new, more efficient way to print and store large volume output listings! The 2680A has the capability to print 2:1 or 4:1 replicas of full-size documents (e.g. two or four scaled-down pages printed per physical page). The resolution possible with these reductions is also quite good.

How can these reductions be accomplished? A number of prepared "environment files" are included in the software accompanying your 2680A purchase. These environment files allow you to alter the manner in which your data is formatted on the page. For example, there is an environment file which will cause data to be printed in a 2:1 reduction of a typical line printer formatted page. This file should be present as LP2.ENV2680A.SYS on your system*. The corresponding environment file that handles 4:1 line printer reductions is LP4, located in the same group and account.

Let us assume that you wish to print a large spoolfile with a devicefileid of #O12, and wish to store it more efficiently by using 2:1 reduction to cut paper requirements in half. When the spoolfile #O12 is READY on the system to be printed, execute the following steps:

```
:FILE E;DEV=2680devclass;ENV=LP2.ENV2680A.SYS
:RUN SPOOK.PUB.SYS
>COPY #O12;ALL,*E
>EXIT
```

Finally adjust the output priorities to let the new spoolfile print. You should see the "compressed" results appearing under your 2680A viewing screen!

Different ways of formatting your data are possible with other HP-supplied environment files. For example, the above 2:1 reduction sequence could be done so that the printed output appeared in "document" form, where the output format uses a Roman font and proportional spacing.

The environment files all reside in the ENV2680A group of the SYS account, and may be used by starting with a basic spoolfile and following the sequence described above. However, if you would rather programmatically create a spoolfile with an environment file included, this can be done by referencing an environment file directly through the DEV parameter of the FOPEN intrinsic. This procedure is described in detail in the MPE Intrinsics manual. Further, if you wish to create your own custom-made environment files for both programmatic and interactive use, please refer to the Interactive Formatting System (IFS/3000) Reference Manual, part number 36580-900001.

* The "2" suffix on the file name indicates a 2:1 reduction of the basic environment file type. The stem,"LP", indicates the basic environment type.

The HP 2680A Translator

by Greg Slansky, Boise Division

The HP 2680A Laser Printing System (LPS) provides a unique set of features for formatting and displaying printed output. Access to these features is accomplished programmatically via several new HP3000 intrinsics documented elsewhere in this issue.

Through Translator commands embedded in EDIT/3000 or TDP/3000 files, the user may exercise all LPS features, such as character set selection, dynamic environment modification, etc. A detailed user's manual and source code for the Translator is available in the HP3000 Contributed Library.

Editor's Note: The 2680A Translator is not an HP supported program.

2680A Software Configuration

by Stephanie Littell, Computer Systems Division

When the new software configuration is done on your system, the 2680A Page Printer should be configured as a line printer, type 32. A new subtype has been added to MPE in order to differentiate the 2680A from other printers. This is subtype 8, which must also be specified in the software configuration.

The 2680A printer requires a new driver, HIOPPRT0, to be added to the software configuration. With the exception of the "Initially Spooled" category (covered elsewhere in this issue), all other configuration parameters are the same as those given in the System Manager/System Supervisor manual for other printers.

D-IT Data Communication Requirements

by Gary Atkins, Boise Division

As discussed elsewhere in this issue, the MPE File System and Spooler have been enhanced to support the addition of the 2680A Laser Printing System. As a result, the "D" Installation Tape must be installed on all systems in a 3000 network if 2680A environment files are to be transferred between nodes.

Spooler Powerfail Recovery Procedures for the 2680A Laser Page Printer

by Bill Tyler, Computer Systems Division

Three different cases occur when the 2680 has a power failure.

1. If a print job is in progress, and the system is able to recover, these messages will be displayed on the system console:

   LDEV ldn HAS BEEN RESET OR POWERFAIL, CHECK PAPER ALIGNMENT AND VFC IF NECESSARY. I/O STATUS %213

   LDEV ldn; MESSAGE BEING DISPLAYED ON THE HP2680 CONSOLE

   LDEV ldn NOT READY

   The 2680 will display the message

   JOB ACTIVE PWR FAIL.

   The operator should check, and if necessry, adjust top-of-form alignment on the 2680. Pressing RUN on the 2680 console will then allow the print job to continue. If a page was partially printed prior to the power fail, the partial page will be entirely reprinted.

2. If a print job is in progress, and the system is unable to recover automatically, the messages

   LDEV ldn POWER FAIL ABORT. I/O STATUS %63

   SP# ldn/STOPPED, SPOOLEE I/O ERROR

   will be displayed on the console. To continue output on the 2680, the operator should enter a STARTSPOOL command. Printing will begin with the next available spoolfile. (The file printing at the time of the power failure will be placed at the end of the queue of spoolfiles for that device. If desired, the operator can alter spoolfile priorities with the ALTSPOOLFILE command prior to restarting spooling.)

3. If no print job was in progress, the 2680 will display the message

   HP2680 PAGE PRINTER

   The operator may simply press RUN on the 2680 to ready it for the next print job.

MPE Spooler Changes for the 2680A

by Stephanie Littell, Computer Systems Division

New enhancements have been made to the MPE Spooler to allow for special features of the 2680A Page Printer. These spooler changes mandate that the 2680A be configured as "Initially Spooled," and also continue to remain spooled while the system is in operation. It is NOT possible to run the printer "hot", e.g. one cannot output to the Page Printer if spooling has been stopped. Again, this is due to the special data structures that the 2680A expects to see formatted by the MPE spooler. Spooler interactions with other printers have not been affected.

For this reason, if a 2680A is present at your installation, your system manager may wish to start the spooling process in two different ways on the 2680A:

```
:STARTSPOOL ldn
:STARTSPOOL 2680devclass
```

By doing this, users can still continue to create and print spoolfiles if a Spoolee I/O Error should occur, and one spooling process thus becomes disabled.

Another spooler enhancement was made in the area of 2680A error reporting. Within the 2680A, there is a microcode processor that actually "parses" the spoolfile to be printed. If a programmatic error is detected, an error file is printed at the end of the print job just before the job trailer. This error file will contain a statistical snapshot of the printer environment at the time of the error. The information provided may be useful in debugging program applications. For example,

```
          ***   ERROR LOG FOR LDEV = 156 DFID = #0220 ***
          ***                                  ***
          ***  ENVIRONMENTAL STATISTICS        ***
          ***                                  ***
BUFFER = 16 BLOCKS  BUCKETS AVAILABLE = 11964 BUCKETS USED = 950
MEMORY = 256K WORDS
2 VFC'S  0 FORMS  2 CHARACTER SETS  1 ACTIVE LOGICAL PAGES
CHARACTER SETS = 4468 WORDS    FORMS = 0 WORDS     VFCS = 132 WORDS
PAGE LENGTH = 8.50 INCHES    ( 21.59 CM)
PAGE WIDTH  = 11.0 INCHES    ( 27.94 CM)
FORMS CLIPPED = NO  DATA TRUNC = NO  CHARS CLIPPED = 2386
TOTAL USER MEMORY = 4604 WORDS    DATE CODE = 2124
TOTAL PHYSICAL PAGES PRINTED = 38
```

This error file was printed after a print job because the character set used was too large to accomodate all characters within the spoolfile's page form boundaries. This can be seen by noting that the error file reports a "characters clipped" value. Thus to correct this situation, a new environment file that uses a smaller character set size, or enlarges the logical page boundaries should be used.

Note that the error file reports only "programmatic" errors concerned with the spoolfile. Hardware errors are reported through other means, for example, through the I/O error type records available in System Logging.

Enhanced Laser Printer Support with DSN/MRJE and DSN/RJE

by Jitendra Singh, Information Networks Division

The DSN/MRJE user can now have job output printed on special forms on the HP 2680 Laser Page Printer. This is done by including the necessary JCL forms control statements in the job stream. The special forms do not have to be mounted by an operator; the specification in the environment file will automatically be used to print the forms and to format the output.

In order to use the environment file feature, one must do the following:

- Configure the 2680 Page Printer to be in device class "LPS".
- Build the environment files in the ENV.HP2680 group and account. The environment file name must be no longer than four characters.
- Enter the name of the environment file in the forms subparameter of the SYSOUT parameter in your job's JCL.

When your output is received from the host, DSN/MRJE will automatically use the environment file named in the forms subparameter to print the data.

This enhancement to DSN/MRJE will not affect users who are not using the 2680 Page Printer.

For a more complete description of this enhancement, refer to the DSN/MRJE Reference Mnual (32192-90001), 3rd Edition, with the new Update #1.

The DSN/RJE product will also allow the use of environment files for printing special forms, but requires an MPE file equation for specifying the file.

MRJE and RJE Performance on the HP 2680A Laser Printing System

by Von Hansen and Larry Goldman, Boise Division

INTRODUCTION

Since the introduction of the 2680A Laser Printing System, many users have expressed interest in printing output from IBM host computers to the 2680A. Testing has been done using DSN/RJE and DSN/MRJE on the HP3000 to show the performance which can be expected for output-only usage.

This article will present test results, explanations, and conclusions about the performance of the HP system in transferring files from IBM host systems to be printed on the HP 2680A Laser Printing System.

CONSIDERATIONS

The primary measures of performance in this environment are the number of pages transferred per minute and the resulting HP3000 CPU overhead.

The things which we believed would influence the performance include:

* speed of the data communications line
* data communications block size
* amount of printable data per page
* the position of printable data in a print line
* model of HP3000 CPU used
* amount of main memory
* number of master discs
* number of data communication links

TEST PROCEDURE

The testing was done using dial-up 4800 and leased 9600 bps communications lines; 400 character blocks were used for 4800 bps; both 400 and 2000 character blocks were used for 9600 bps; various print datasets were constructed and transmitted; HP3000 Series 33 and 44 CPUs were utilized; 512k and 1024k memory sizes were checked; both one and two master disc environments were tested.

The host system was an Amdahl 470-V8 running JES2 for MVS under VM/370. The communications line was connected to the host via an IBM 3705 communications controller. The HP3000 ran RJE or MRJE under the MPE-IV operating system. The communications line was connected to the HP3000 via the INP.

The number of pages per minute was measured by dividing the number of pages transmitted in a single file by the elapsed time required to transmit the file. The elapsed time was determined by examining the time stamps on the header and trailer pages from the host system.

The CPU utilization was measured as the average percentage CPU utilization of all processes on the HP3000 over the time the file was being transmitted from the host to the HP3000. The OPT/3000 performance tool was used to take the measurements. There were no other users or activity on the HP3000 during testing.

TEST RESULTS

See the appendix for test data, graphs, and examples of the printed output.

There are some notable results of this testing.

1. The CPU model does not significantly affect the number of pages transmitted per minute.

2. The number of printable characters per page has the most significant influence on pages transmitted per minute.

3. Increasing the blocksize of data transmission (in this from 400 to 2000 bytes) increases the pages per minute by 30 to 40 percent. In most realistic cases, the increased blocksize caused a slight decrease in CPU utilization.

4. CPU utilization of a Series 44 is approximately one-third that of a Series 30 in running MRJE. Data on RJE is not available.

5. The CPU utilization increases as the number of printable characters per page decreases. This is because RJE and MRJE are using lots of CPU time decompressing blanks which the host eliminated. This elimination of blanks by the host is what causes more pages per minute to be transmitted at a given line speed.

6. The number of master discs and the amount of memory (over 512k bytes) does not significantly affect performance. This is not to say that these resources would not be important if other applications were running.

7. RJE 3780 results in a similar number of pages per minute as MRJE (400 byte blocks) but with lower CPU utilization.

8. RJE 3780 results in less pages per minute than MRJE (2000 byte blocks) but again with lower CPU utilization.

There are other factors to consider when evaluating this data:

1. The quality of the communications line affects performance. Low quality lines result in data retransmission which causes lower performance. The 9600 bps line in this test had C1 conditioning.

2. The responsiveness of the host will affect performance. The model of host CPU is not the issue, but rather the load on that CPU and on the communications controller.

3. RJE is supported at 19.2k bps. We do not yet have comparison test cases at 19.2.

4. We measured one file output at a time. In normal operation, a user will get many files each day. With RJE, a human operator must issue an RJE command to cause transmission of each file. MRJE causes each file to be transmitted automatically, therefore overall throughput is likely to be greater with MRJE.

5. MRJE (on the D-MIT) has support for automatically selecting environment files. RJE utilizes environment files only when the operator preselects the environment in a :FILE command.

6. The position of print on a line is significant for lines which have very little printable data. The further to the right that the printable data starts, the more decompression required, and consequently, more CPU time utilized.

7. Multiple communications lines should improve total throughput on a Series 44 since CPU utilization for a single line is low.

There are some other tests which are needed but have not been completed. We intend to complete these tests in the future.

1. RJE using 19.2k bps on Series 33 and 44.

2. RJE using 9600 bps on Series 44.

3. Multiple communications lines.

CONCLUSIONS

During a given period of time, neither RJE nor MRJE can transmit enough data (of realistic types) to keep the HP 2680A LPS printing at 45 pages per minute. However, over long periods of time enough data can be transferred and stored to keep the 2680A busy for much of it's print duty cycle. Also, many jobs will use the multicopy feature of the 2680A which requires only one copy to be sent from the host.

The best HP system configuration for customer satisfaction is a Series 44 running MRJE using 2000 byte blocks at 9600 bps.

APPENDIX A : TEST DATA

Test #1:

Configuration: MRJE, 9600 baud, 400 byte blocks, series 33, 1024k memory size, MPE IV.

Data set	character /page	pages/test	transmission time	pages/min	%CPU
MRJTEST2	2501	110	8:03	13.66	49
MRJTEST3	2440	110	7:31	14.63	51
MRJTEST4	1240	150	6:07	24.52	68
MRJTEST5	7920	80	14:05	5.68	41
MRJTEST6	588	150	7:25	20.22	60
MRJTEST7	588	150	4:45	31.58	67
MRJTEST8	588	150	6:00	25.00	59
MRJTEST9	588	150	5:23	27.86	71

TEST #2:

Configuration: MRJE, 9600 baud, 2000 byte block, series 33, 1024k memory size, MPE IV.

Data set	character /page	pages/test	transmission time	pages/min	%CPU
MRJTEST2	2501	110	5:39	19.47	45
MRJTEST3	2440	110	5:34	19.76	47
MRJTEST4	1240	150	4:37	32.49	69
MRJTEST5	7920	80	10:17	7.78	30
MRJTEST6	588	150	5:06	29.41	73
MRJTEST7	588	150	3:32	42.45	76
MRJTEST8	588	150	4:06	36.59	73
MRJTEST9	588	150	4:57	30.30	70

TEST #3:

Configuration: RJE 3780, 9600 baud, series 33, MPE IV, 1024k memory size.

Data set	character /page	pages/test	transmission time	pages/min	%CPU
MRJTEST2	2501	110	6:46	16.26	31
MRJTEST3	2440	110	6:38	16.58	32
MRJTEST4	1240	150	5:25	27.69	42
MRJTEST5	7920	80	12:42	6.29	25
MRJTEST6	588	150	3:02	49.45	75
MRJTEST7	588	150	2:44	54.88	65
MRJTEST8	588	150	2:47	53.89	67
MRJTEST9	588	150	2:50	52.94	60

TEST #4:

Configuration: MRJE, 9600 baud, 2000 byte block, series 44, 1024k memory size, MPE IV.

Data set	character /page	pages/test	transmission time	pages/min	%CPU
MRJTEST2	2501	110	5:30	20.00	14
MRJTEST3	2440	110	5:27	20.18	14
MRJTEST4	1240	150	4:27	33.70	21
MRJTEST5	7920	80	10:11	7.85	9
MRJTEST6	588	150	2:37	57.32	28
MRJTEST7	588	150	1:59	75.63	29
MRJTEST8	588	150	2:15	66.66	30
MRJTEST9	588	150	2:43	55.21	37

TEST #5:

Configuration: MRJE, 9600 baud, 400 byte block, series 44, 512k memory size, MPE IV.

Data set	character /page	pages/test	transmission time	pages/min	%CPU
MRJTEST2	2501	110	xxxx	15.50	18
MRJTEST3	2440	110	xxxx	15.70	18
MRJTEST4	1240	150	xxxx	27.80	24
MRJTEST5	7920	80	xxxx	6.35	16

TEST #6:

Configuration: MRJE, 4800 baud, 400 byte blocks, series 44, 512k memory size, MPE IV.

Data set	character /page	pages/test	transmission time	pages/min	%CPU
MRJTEST2	2501	110	xxxx	6.9	9
MRJTEST3	2440	110	xxxx	7.3	9
MRJTEST4	1240	150	xxxx	12.8	12
MRJTEST5	7920	80	xxxx	2.7	8

TEST #7:

Configuration: MRJE, 9600 baud, 400 byte block, series 30, MPE 4, 512k memory size.

Data set	character /page	pages/test	transmission time	pages/min	%CPU
MRJTEST2	2501	110	xxxx	13.41	53
MRJTEST3	2440	110	xxxx	14.93	57
MRJTEST4	1240	150	xxxx	25.55	74
MRJTEST5	7920	80	xxxx	5.88	48

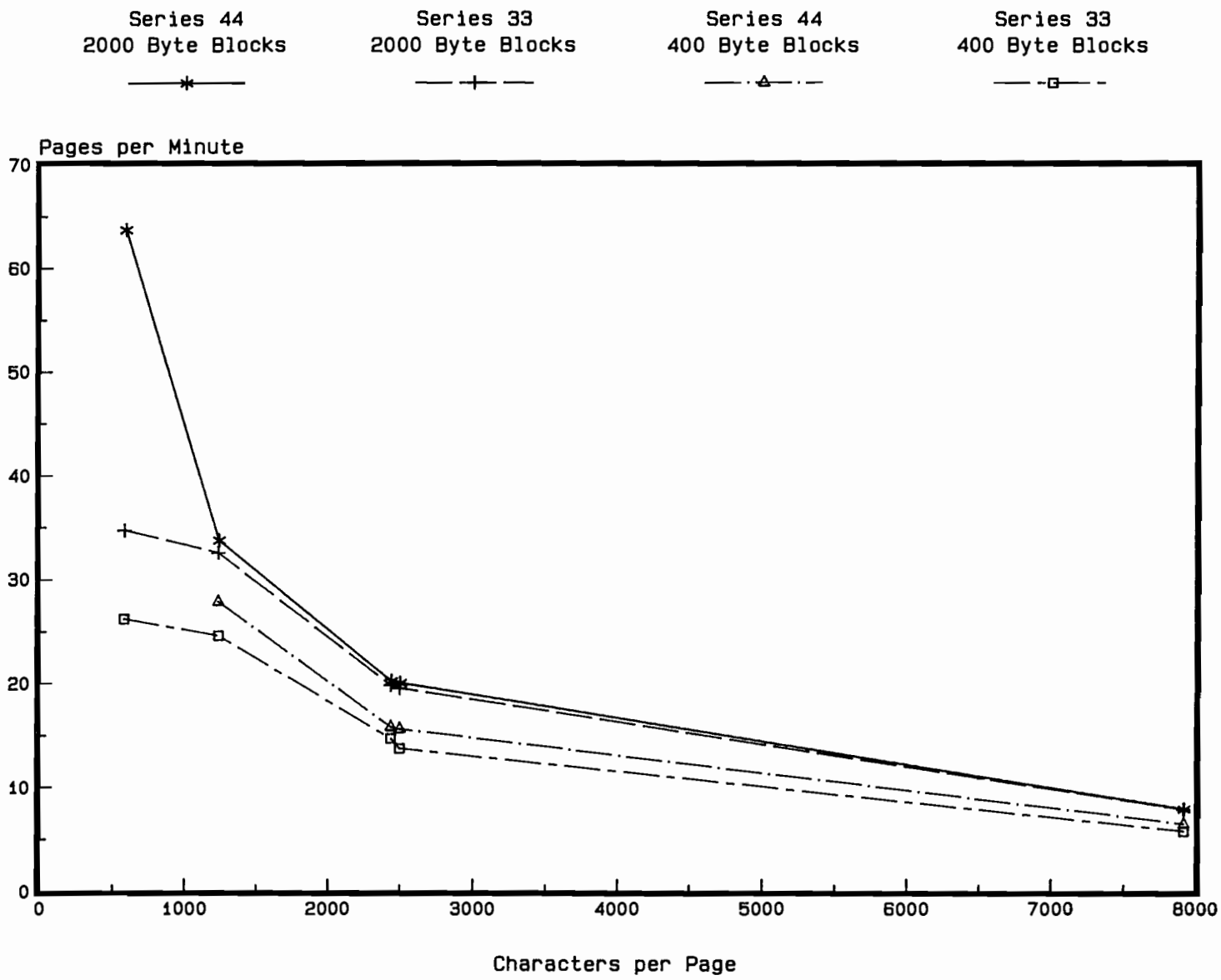
TEST #8:

Configuration: MRJE, 4800 baud, 400 byte blocks, series 30, 512k memory size, MPE IV.

Data set	character /page	pages/test	transmission time	pages/min	%CPU
MRJTEST2	2501	110	xxxx	6.88	xx
MRJTEST3	2440	110	xxxx	7.40	30
MRJTEST4	1240	150	xxxx	12.96	40
MRJTEST5	7920	80	xxxx	2.79	23

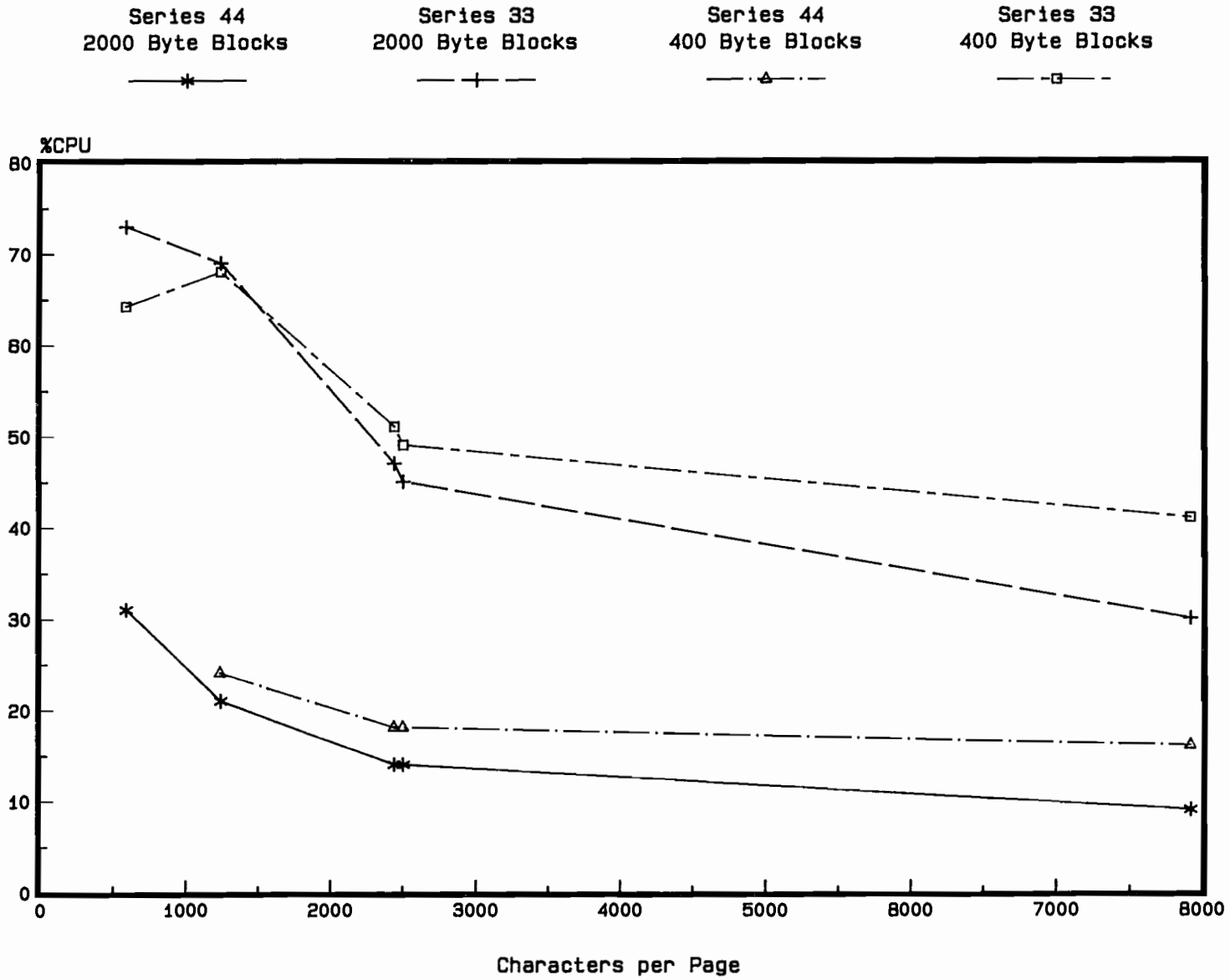
MRJE DATA THROUGHPUT

9600 BAUD

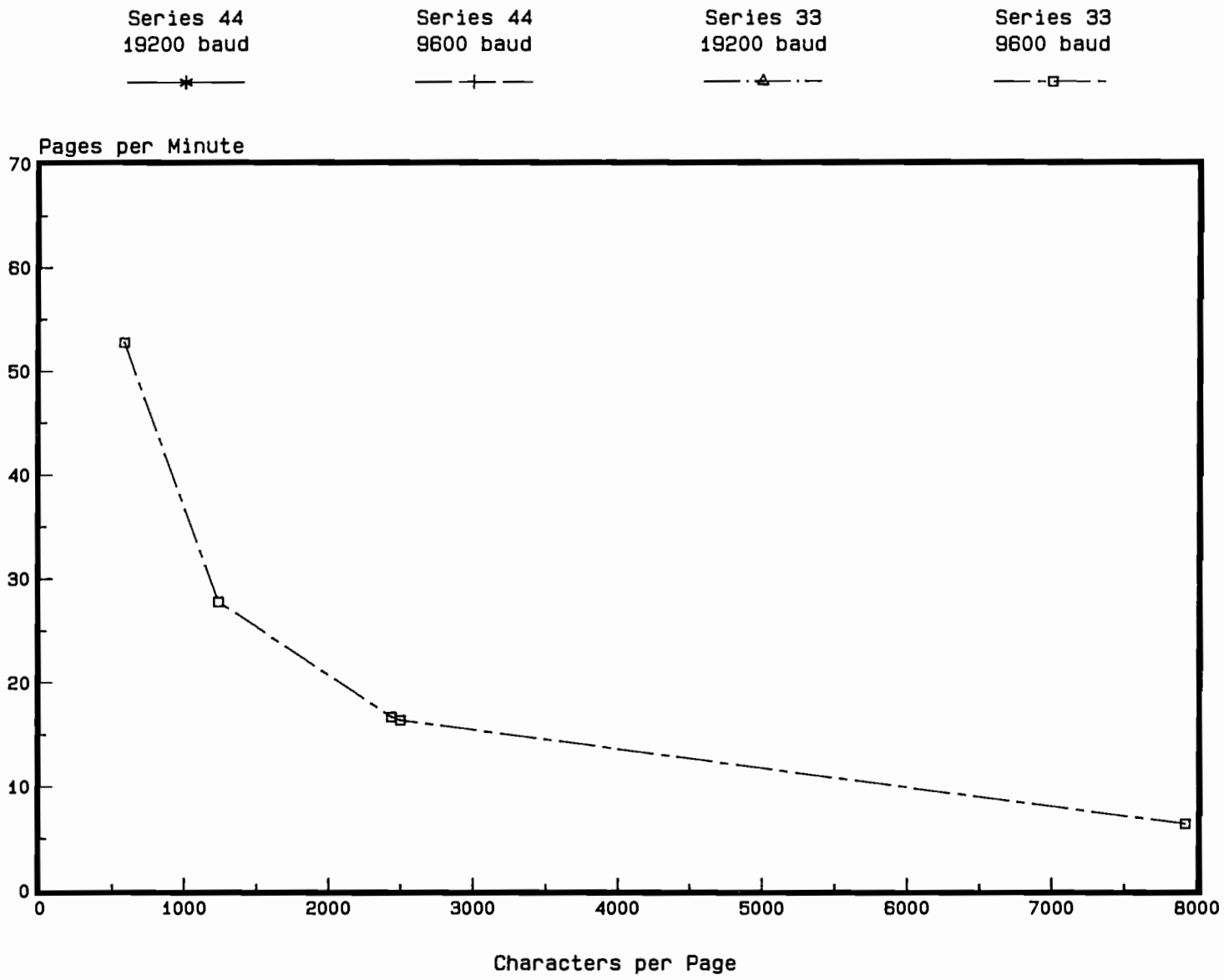


MRJE CPU UTILIZATION

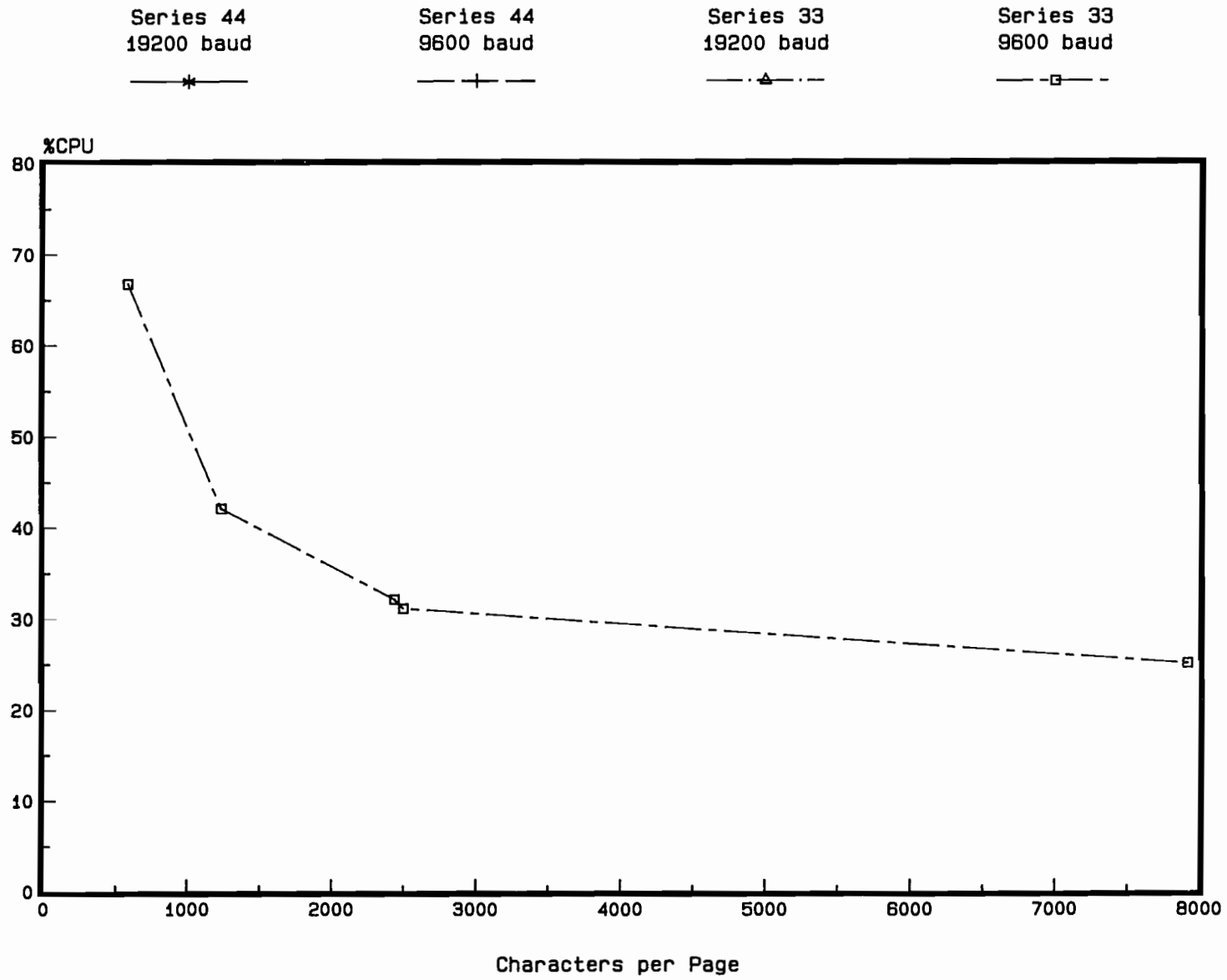
9600 BAUD



RJE DATA THROUGHPUT



RJE CPU UTILIZATION



```
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567848
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 1
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 2
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 3
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 4
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 5
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 6
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 7
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 8
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   12345678 9
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567810
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567811
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567812
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567813
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567814
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567815
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567816
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567817
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567818
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567819
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567820
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567821
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567822
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567823
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567824
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567825
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567826
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567827
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567828
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567829
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567830
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567831
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567832
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567833
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567834
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567835
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567836
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567837
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567838
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567839
ABCDE   IJKLMNOP   UVWXYZ1234567890   4321   CDEFGHI   Z!"#$%   0=,.;   1234567840
```

DATA SET: MRJTEST2

61 CHAR/LINE
41 LINES/PAGE
2501 CHAR/PAGE

ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567841
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	12345678 2
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	12345678 3
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	12345678 4
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	12345678 5
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	12345678 6
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	12345678 7
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	12345678 8
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	12345678 9
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567810
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567811
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567812
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567813
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567814
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567815
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567816
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567817
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567818
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567819
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567820
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567821
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567822
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567823
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567824
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567825
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567826
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567827
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567828
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567829
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567830
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567831
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567832
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567833
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567834
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567835
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567836
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567837
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567838
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567839
ABCDE	IJKLMNOP	UVWXYZ1234567890	4321	CDEFGHI	Z!"#$%	0=,.;	1234567840

DATA SET: MRJTEST3

61 CHAR/LINE
40 LINES/PAGE
2440 CHAR/PAGE

ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67819
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	678 2
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	678 3
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	678 4
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	678 5
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	678 6
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	678 7
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	678 8
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	678 9
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67810
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67811
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67812
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67813
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67814
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67815
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67816
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67817
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67818
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67819
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67820
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67821
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67822
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67823
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67824
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67825
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67826
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67827
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67828
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67829
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67830
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67831
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67832
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67833
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67834
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67835
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67836
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67837
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67838
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67839
ABCDE	IJK	UVWXYZ	4321	CD	Z!"#$%	67840

DATA SET: MRJTEST4

31 CHAR/LINE
40 LINES/PAGE
1240 CHAR/PAGE

ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567823
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@12345678 2
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@12345678 3
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@12345678 4
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@12345678 5
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@12345678 6
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@12345678 7
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@12345678 8
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@12345678 9
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567810
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567811
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567812
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567813
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567814
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567815
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567816
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567817
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567818
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567819
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567820
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567821
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567822
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567823
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567824
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567825
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567826
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567827
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567828
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567829
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567830
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567831
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567832
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567833
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567834
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567835
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567836
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567837
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567838
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567839
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567840
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567841
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567842
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567843
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567844
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567845
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567846
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567847
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567848
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567849
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567850
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567851
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567852
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567853
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567854
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567855
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567856
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567857
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567858
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567859
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890098765432 1ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ1234567!"#$%&'()0=,.;:/@1234567860

DATA SET: MRJTEST5
132 CHAR/LINE
60 LINES/PAGE
7920 CHAR/PAGE

```
123456789011
1234567890 2
1234567890 3
1234567890 4
1234567890 5
1234567890 6
1234567890 7
1234567890 8
1234567890 9
123456789010
123456789011
123456789012
123456789013
123456789014
123456789015
123456789016
123456789017
123456789018
123456789019
123456789020
123456789021
123456789022
123456789023
123456789024
123456789025
123456789026
123456789027
123456789028
123456789029
123456789030
123456789031
123456789032
123456789033
123456789034
123456789035
123456789036
123456789037
123456789038
123456789039
123456789040
123456789041
123456789042
123456789043
123456789044
123456789045
123456789046
123456789047
123456789048
123456789049
```

DATA SET: MRJTEST6

12 CHAR/LINE
49 LINES/PAGE
588 CHAR/PAGE

```
1234567890 1
1234567890 2
1234567890 3
1234567890 4
1234567890 5
1234567890 6
1234567890 7
1234567890 8
1234567890 9
123456789010
123456789011
123456789012
123456789013
123456789014
123456789015
123456789016
123456789017
123456789018
123456789019
123456789020
123456789021
123456789022
123456789023
123456789024
123456789025
123456789026
123456789027
123456789028
123456789029
123456789030
123456789031
123456789032
123456789033
123456789034
123456789035
123456789036
123456789037
123456789038
123456789039
123456789040
123456789041
123456789042
123456789043
123456789044
123456789045
123456789046
123456789047
123456789048
123456789049
```


DATA SET: MRJTEST7

12 CHAR/LINE
49 LINES/PAGE
599 CHAR/PAGE

```
1234567890 1
1234567890 2
1234567890 3
1234567890 4
1234567890 5
1234567890 6
1234567890 7
1234567890 8
1234567890 9
1234567890 10
1234567890 11
1234567890 12
1234567890 13
1234567890 14
1234567890 15
1234567890 16
1234567890 17
1234567890 18
1234567890 19
1234567890 20
1234567890 21
1234567890 22
1234567890 23
1234567890 24
1234567890 25
1234567890 26
1234567890 27
1234567890 28
1234567890 29
1234567890 30
1234567890 31
1234567890 32
1234567890 33
1234567890 34
1234567890 35
1234567890 36
1234567890 37
1234567890 38
1234567890 39
1234567890 40
1234567890 41
1234567890 42
1234567890 43
1234567890 44
1234567890 45
1234567890 46
1234567890 47
1234567890 48
1234567890 49
```


DATA SET: MRJTEST8

12 CHAR/LINE
49 LINES/PAGE
588 CHAR/PAGE

12	3456	7890 1
12	3456	7890 2
12	3456	7890 3
12	3456	7890 4
12	3456	7890 5
12	3456	7890 6
12	3456	7890 7
12	3456	7890 8
12	3456	7890 9
12	3456	789010
12	3456	789011
12	3456	789012
12	3456	789013
12	3456	789014
12	3456	789015
12	3456	789016
12	3456	789017
12	3456	789018
12	3456	789019
12	3456	789020
12	3456	789021
12	3456	789022
12	3456	789023
12	3456	789024
12	3456	789025
12	3456	789026
12	3456	789027
12	3456	789028
12	3456	789029
12	3456	789030
12	3456	789031
12	3456	789032
12	3456	789033
12	3456	789034
12	3456	789035
12	3456	789036
12	3456	789037
12	3456	789038
12	3456	789039
12	3456	789040
12	3456	789041
12	3456	789042
12	3456	789043
12	3456	789044
12	3456	789045
12	3456	789046
12	3456	789047
12	3456	789048
12	3456	789049

DATA SET: MRJTEST9

12 CHAR/LINE
49 LINES/PAGE
588 CHAR/PAGE

Introducing the On-Line Performance TOOL/3000 Performance Measurement Package

by Robin Rakusin, Computer Systems Division

The On-line Performance Tool/3000 (OPT/3000) Performance Measurement Package consists of two interdependent products:

- OPT/3000 Software
- OPT/3000 System Performance Training Course

This Package is designed to help HP 3000 users gather and utilize performance data for system management, capacity planning, and application development activities.

This article will first discuss OPT/3000 Software, including its features, information included in the 6 Display Contexts, terminal requirements, and availability on HP 3000 systems. The article will then briefly discuss the OPT/3000 System Performance Training Course. This Course is designed to teach users how to use OPT/3000 software to generate performance related data and how to interpret the results.

OPT/3000 Software

On-line Performance Tool/3000 (Product 32238A) is an interactive performance measurement software product for the HP 3000 that provides information to the system analyst. OPT/3000 can be used to characterize the current system workload, CPU utilization, memory management activity, I/O traffic, program and process activity, and system table usage to help the user isolate bottlenecks and improve system performance. Performance data provided by OPT/3000 is continuously updated at regular intervals and can be presented in charts, graphic displays, or summary reports. Information can be displayed on an HP 264x series terminal and a hard copy of any display can be generated on a line printer with a single keystroke. Although OPT/3000 is primarily designed for interactive use, it can be executed in batch mode to collect snapshots of system activity over a period of time.

Features

- Interactive Terminal Reporting of Performance-Related Data: OPT/3000 provides summary and detailed CPU, memory, I/O, and process information in dynamically updated terminal displays. The displays are updated automatically or by simply pressing the return key.

- Graphical Presentation of Information: OPT/3000 utilizes the features of the HP 264x series of terminals to generate displays with a graphical format. The terminal video enhancements used include blinking, inverse video, underlining, and half-bright. The line drawing character set and the cursor addressing capabilities are also used.

- Multiple Display Levels: The displays associated with a particular context are layered such that each level provides the user with more detailed information than the previous level.

- Summary Reporting of Resource Usage: Summary information about system activity can be generated interactively or in batch mode. These reports can be used to provide data for capacity planning activities.

- Hard Copy Capability: A hard copy of any display can be generated on the line printer with a single keystroke.

- Logging Capability: The information used to generate the summary reports can also be logged into a disc file. This data can be accessed by user programs for reformatting and reporting at a later time.

- Low Overhead: The performance impact of OPT/3000 running on an HP 3000 is low, varying from one to three percent of available CPU time. Multiple users of OPT/3000 all access the same set of performance information maintained by the MPE Measurement Interface Facility. As a result, multiple users of OPT/3000 on a system can be executing simultaneously with little additional impact on system performance.

- Extensive On-Line Help Facility: With this integrated facility, documentation explaining any command or context display can be quickly and easily displayed. In many cases, interpretation guidelines are provided to aid in the identification of performance problems.

Information Included in the 6 Display Contexts

OPT/3000 can generate 23 unique displays containing system performance information. These displays are grouped into six categories called display contexts. Each context is associated with a different type of system resource. These six Display Contexts are:

- Global
- Memory
- CPU-Memory Manager
- I/O
- Process
- System Tables

Within each context, displays are available at successively greater levels of detail. This structure allows the user to progress from summary level information to more detailed information as required.

Global Context

The Global Context is automatically entered upon execution of OPT/3000. This context consists of two displays that provide summary level information describing CPU usage, memory utilization, disc I/O rates, and process activity. These two displays can be used to quickly determine potential problem areas within the system prior to accessing more detailed displays in the other contexts. These detailed displays can then be used to isolate and verify potential performance problems at a much more detailed level. The Global Context can also be used to monitor general system activity in order to detect fluctuations in resource usage.

Memory Context

The Memory Context consists of eight displays that provide information related to the usage of main memory and segment sizes. Three of the displays describe the current contents of memory and the remaining five displays consist of histograms depicting distributions of segment sizes or free areas in memory. The highest level display allows the user to determine the current percentages of main memory that contain code segments, stacks, and extra data segments. The user is also able to generate an image of the entire contents of main memory, or just a single bank of memory, that indicates the type and approximate size of each segment. The highest level histogram display depicts separate distributions for code, stack, and extra data segments. The remaining four histogram displays generate higher resolution segment distributions as well as a distribution of the free areas in memory.

CPU-Memory Manager Context

The CPU-Memory Manager Context consists of three displays that provide information related to CPU usage and memory management activity. The CPU information describes the percentage of time the CPU is in various states and the rate at which processes are being allowed to execute in the CPU. These various states include CPU time for execution, memory management, overhead processing, waiting, and CPU idle time. Event rate information is reported for memory management activity, including memory allocations, memory management disc I/Os, and segments released from memory. Information is also available regarding how the memory manager satisfies requests for absent segments.

I/O Context

The I/O Context consists of four displays that provide I/O completion rates for disc drives, line printers, and magnetic tape drives. The highest level display describes the I/O completion rate per second for each type of device for both the current and overall intervals. The remaining three displays provide more detailed device information such as the completion rates for read, write, and control operations on each individual device.

Process Context

The Process Context consists of four displays that provide information on process and program activity on the system. The highest level display describes all active or allocated programs including the program file name, the size of the program file in words, the number of segments in the program, the number of current users of the program, and limited working set information. The remaining displays provide more detailed information about the process including the user name and account, the process identification number (PIN), and stack space utilization. Additional information on the various states of all processes is provided. These various states include the number of processes waiting for I/O and the number of processes in the dispatch queue.

System Tables Context

The System Table Context consists of two displays that provide the current and maximum utilization of configurable system tables. One display describes the current and maximum utilizations in a graphical format, while the other display provides more detailed information in a tabular format. This information includes the configured number of entries, entry size, and current and maximum table utilizations.

Terminal Requirements for Running OPT/3000

HP 2645A, HP 2647A, or HP 2648A terminals with the following display features are required for OPT/3000:

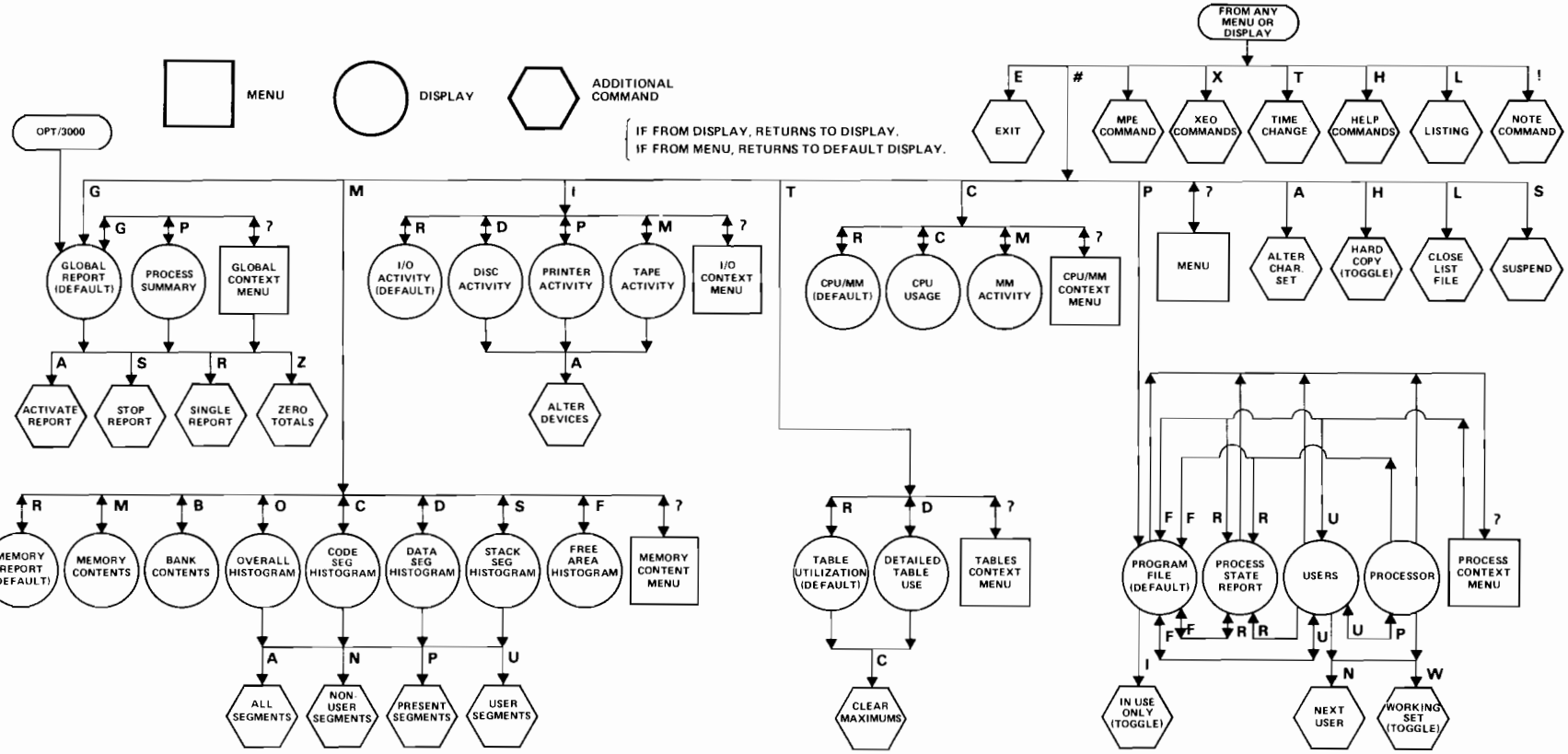
- Line Drawing Set
- Inverse video, half-bright, underline, and blinking
- Upper and lower case letters
- Minimum of 8 Kbytes display memory (12 Kbytes for multipoint if the datacomm buffer is configured for 4 K characters, and for an HP 3000 Series 44 with more than 2 Mbytes of memory.)

OPT/3000 is not supported on other HP 262x and HP 264x series terminals without the features and memory requirements listed above. The limited display capabilities of these terminals causes a substantial reduction in the clarity of OPT/3000 displays. Non-HP terminals are not supported by OPT/3000. Hard copy capability on multipoint configurations is limited to the Memory Context displays.

Availability of OPT/3000 on HP 3000 Systems

OPT/3000 is available for installation on every HP 3000 Series II, III, 30,33, and 44 running MPE IV (C Installation Tape) or later versions of the HP 3000 Multiprogramming Executive Operating System. An OPT/3000 Reference Manual (32238-90001) is supplied with OPT/3000 Software (Product 32238A). This Reference Manual gives detailed information on the operation of and information presented by OPT/3000. A convenient OPT/3000 Pocket Card (32238-90002) is also included to facilitate the operation of OPT/3000. The roadmap and usage guidelines included on the Card are illustrated on the following pages. Note that OPT/3000 is invoked by:

```
RUN OPT.PUB.SYS
```



Enlargement of OPT/3000 Pocket Card

Control Operation Commands

# ?	Menu display for control operation commands

To Enter New Context

# C	CPU/Memory Manager Context
# G	Global Context (defaults to Global Report)
# I	I/O Context
# M	Memory Context
# P	Process Context (defaults to Program File)
# T	Systems Tables Context

Note: The following commands return user to calling display or to the context default display if called from a menu display.

# A	Alter line drawing character set position
# H	Continuous hard copy generation toggle
# L	Close listing file
# S	Temporarily suspend display updating

Display Control Commands

E	Exit
H	Access on-line HELP facility
L	Hard copy list of current display
T	Change update time interval
X	Define sequence of commands
?	Menu display command for current context
:	Execute MPE commands
!	Generate note

Batch Access Control	default
Minutes between summary reports?	(60)
Reports between closing of report file?	(8)
File closings before termination?	(3)
Name of log file?	(none)
Summary report identification?	(none)

Enlargement of OPT/3000 Pocket Card

OPT/3000 System Performance Training Course

The OPT/3000 System Performance Training Course (Product 22809A) consists of ten days of training at an HP technical center conducted by an HP Performance Specialist. The Course covers the internals of the HP 3000 (MPE) Multiprogramming Executive Operating System and system performance analysis on the HP 3000. Details on the operating system, including process management, memory management, scheduling/dispatching, and I/O, are covered in the MPE internals section of the Course. The interrelationships of system performance variables, an OPT/3000 overview, and case studies are covered in the second half of the Course. This Course is required for one person with every initial installation of OPT/3000 (Product 32238A) Software. It is also recommended for anyone interested in a course on MPE internals and HP 3000 system performance analysis. The number of students who may be included in the class is limited to two students per available terminal.

Availability

The OPT/3000 System Performance Training Course will be available in HP technical centers beginning September 1981. The Course is ten days long, is taught by HP Performance Specialists, and includes approximately 60% lab time to provide each student with extensive hands-on experience.

Editor's Note: The following article is a reprint of an earlier press release.

High-Speed Half-Inch Tape Drive for HP Computers

A new, 6250/1600 BPI, ANSI-format, half-inch tape subsystem for Hewlett-Packard computers now makes it possible for HP 3000 users to transfer data from a disc in a steady stream at 320 kilobytes per second, while retaining the traditional, logical file sequence on the tape. In the past, non-stop streaming was possible only by accepting files on tape in whatever sequence they occur on the disc. By combining a new data buffering scheme with new software, the ability to read and write in two ANSI standard tape formats is paired with the advantage of high-speed non-stop tape motion. The new HP 7976A tape drive can complete a 400-megabyte disc backup, with files in logical sequence on the tape, in less than 30 minutes.

The HP 7976A is a complete half-inch, nine-track magnetic tape subsystem, with automatic tape-loading and tape-threading. It consists of a tape transport, formatter/controller unit and an HP-IB (IEEE 488) interface controller in an upright cabinet. The transport can read and write at 75 inches per second and rewinds at 250 ips. Vacuum-column tape buffering controls tape tension, and protects the tape medium. The tape subsystem can produce either of two ANSI formats, data 6250 BPI group-coded recorded, or tapes 1600 BPI phase-encoded. The format is program selectable.

Records in Logical Sequence

The HP 7976A and the system software were specifically designed for high data transfer speeds, especially when performing disc file back-ups or restorations. By using both a new buffering scheme in the interface controller, and new software, two commands from the host CPU can be recognized concurrently. One buffer of data is emptied to the tape by the first command, while a second buffer fills with data to be acted upon later. This technique of command queuing eliminates the inter-record starting and stopping common to magnetic tape drives. During a normal back-up operation, data is streamed to the tape, the drive running at constant speed. Using data blocks of 16 kilobytes length, and writing at 6250 BPI, a 400-megabyte disc backup is accomplished in less than 30 minutes. Continuous data transfer speed is 320 kilobytes per second. The streaming mode works equally well with the optional 1600 BPI phase-encoded format, although at a slower data transfer rate. The HP 7976A will read or write in the streaming mode, and will even switch between start/stop operations and streaming mode automatically, depending on the availability of data from the computer system. Onboard error correction is also handled by the HP 7976A with no system processor overhead.

The integral HP-IB (IEEE 488) interface was designed to enhance the performance of the tape subsystem when it is paired with HP 3000 computer systems. The HP 7976A can be interfaced directly to HP 3000s of Series 30, 33, and 44, as well as Series III when used concurrently with the newly introduced HP-IB Interface Module.

Built-in Self Test

A user-operated self-test capability provides fast verification of proper operation. As an added service convenience, the subsystem is fitted with an RS-232 diagnostic port. A qualified technician can plug an HP-85 Personal Professional Computer or an HP 2645A Terminal directly into the unit, and run a complete diagnostic check using in the terminal a cartridge tape designed for this purpose.

7976A Tape Drive Performance and Tape Requirements

by Bill Dalton, Computer Systems Division

The 7976A Tape Drive and system software have been designed to provide significantly improved performance over the 7970E for backup and restore operations (SYSDUMP, STORE, RESTORE, DBSTORE, AND DBRESTORE) to unlabelled tape. (Note that performance will not differ significantly from the 7970E when using labelled tape, and when accessing the 7976A via DS lines.)

Testing on the 7976A has shown performance improvements of 3.5 to 6.5 times that of the 7970E when operated at 6250 BPI. At 1600 BPI, the improvement was 1.5 to 2.0 times that of the 7970E. The exact improvement is determined by both the file size and extent size. Large files and files with large extents show the greatest improvement: extent size is more important than file size.

Operating on an HP 3000 Series 44, the 7976A can backup a 400 megabyte system in 30 to 40 minutes (files with large extents and 16KB records at 6250 BPI). Backup for a 2 gigabyte system (sixteen 7925A Disc Drives) will take less than 2 1/2 to 3 1/2 hours including time for rewinding and changing reels. Additionally, only 14 reels of 2400 foot tape is required, versus 50 reels for the same backup operation on a 7970E.

In order to provide improved performance for user programs, the File System has been modified to increase the total buffer space available for each file from 16KB to 28KB. Maximum block size for the 7976 Tape Drive is 16KB.

Using the 7976A Tape Drive

The 7976A Tape Drive presents the same interface as the 7970E to user programs and commands, with these exceptions:

- the 7976A is much faster
- block size is limited to 16KB for the 7976
- the 7976 can operate at either 1600 or 6250 BPI
- the 7976 writes longer file mark gaps

To allow the user the choice of density, the :FILE command and FOPEN Intrinsic have been enhanced as discussed in the article on File System Enhancements for the D Installation Tape which appears elsewhere in this issue. Default density for the 7976A is 6250 BPI. Note that it is alright to specify density when accessing files on a 7970E, and no error is returned.

It should be noted that in order to be compatible with industry standards, the 7976A creates file mark gaps of 3.6 inches, while the 7970 tape drive writes file mark gaps of .6 inch. Therefore, an application which generates tapes written at 1600 BPI may require more tape on the 7976 than the same application on a 7970 tape drive. It should also be noted that labelled tapes use more file marks than unlabelled tapes.

Configuration

The 7976A is configured identically to the 7970E, with the exceptions of the driver, HIOTAPE1, and the subtype, which is 1 for operator allocation, and 9 for automatic allocation.

New On-Line Diagnostics for HP 2680A and HP 7976A

by Jim Chiochios, Computer Systems Division

With the introduction of the HP 2680A Page Printer and the HP 7976A Magnetic Tape Unit, we are also introducing the ability to test/diagnose these products on-line. These tools are available to anyone with System Manager (SM) or System Supervisor (OP) capabilities.

A new group has been established for on-line diagnostics relating to Input/Output (I/O) devices, which is HP32340 in the Support account. The program files for the diagnostics are:

PD467A - HP 2680A Page Printer

PD470A - HP 7976A Mag Tape Drive

These programs will allow a user or Customer Engineer to diagnose problems or confirm proper operation of these new devices on their system. The 7976A diagnostic (PD470A) will diagnose a tape problem by downloading diagnostic software and routines into the tape drive. These routines will diagnose problems and return appropriate error messages that will aid the Customer Engineer in troubleshooting the tape drive. Whenever possible, the diagnostic will report the probable printed cirucit assembly or assemblies that are causing the failure. (The error messages will use abbreviations relating to functions or printed circuit assemblies that may not be understandable by a user. If failures occur, contact your local service office for assistance.)

The 2680A Page Printer program (PD467A) is a verifier rather than a diagnostic. It has some capability of determining which printed circuit assemblies are defective, but in most cases, it relies on status being returned from the printer and the actual output on paper. This program does use all of the capabilities that are available to a user, and it will demonstrate this as part of the standard test. This program can be used to determine not only that the printer is working correctly, but that a user has set up the environment for the printer correctly.

For systems that are turned off for a period of time, the System Manager may want to utilize these programs to determine proper operation of these devices before they are made available to users. The proper dialogue can be set up in a job file and the system operator can stream this job to determine the condition of these products. An example of a job stream to run the Page Printer Verifier is as follows:

```
:JOB D467A,FIELD.SUPPORT,HP32340;RESTART
:RUN PD467A
N
:TELLOP;PAGE PRINTER TEST COMPLETE
:EOJ
```

More specific information regarding these programs may be obtained from the following manuals:

HP 7976 Magnetic Tape Unit Diagnostic Loader Manual
P/N 30070-90073

HP 2680 Page Printer Verifier Diagnostic Manual
P/N 30070-90074

New Subtypes for Tape Drives

by Carole Johnson, Computer Systems Division

To accommodate the new 7976A Tape Drive, device subtypes for tape drives have been broken into two groups: %0 - %7 for non-automatically allocating drives, and %10 - %17 for mag tape units which can be automatically allocated. (See the System Manager/System Supervisor Reference Manual for details on automatic allocation.)

The low order three bits of the subtype are used to distinguish between the different types of drives. Subtypes %0 and %10 indicate a 7970; %1 and %11 indicate a 7976.

The following table illustrates the different subtypes.

Drive	Subtype	Allocation
7970E	0	Operator
	8 (%10)	Automatic
7976A	1	Operator
	9 (%11)	Automatic

Note that systems currently using subtype 15 for automatic allocation of 7970E drives should be converted to subtype 8 when updating to the "D" IT.

30341A HB-IB Interface Module for Series III

by Dick Ollins, Computer Systems Divison

The new HP-IB Interface Module allows the Series III user to connect some of our newer HP-IB only peripherals. The peripherals supported at first release are the 2680A Laser Page Printer and the 7976A Magnetic Tape Drive. Up to one magnetic tape drive and two laser printers can be connected at the same time.

The hardware interface consists of an I/O bay mounted on the right side of the Series III CPU Bay. Within the I/O bay is a 10 slot card cage and an additional power supply. The card cage has an IMB backplane similar to Series 30 and Series 33 systems, with 10 slots available for printed circuit assemblies. The minimum system consists of four assemblies. The IMB Adaptor is a new assembly which interfaces the IMB Bus to the Series III Port Controller. The GIC is the standard assembly for interfacing HP-IB devices to the IMB Bus. The BIC is the standard assembly used in Series 30 and Series 33 systems. The PROC is a remicrocoded version of the standard assembly used in Series 30 and Series 33 systems. The new microcode is a limited version of the Series 30 and Series 33 microcode, and is primarily intended to run channel program routines.

Series III System Startup with an HP-IB Interface

by Stephanie Littell Computer Systems Division

If you have purchased a 30341A HP-IB Interface for your Series III, there are certain new procedures to follow when starting up the system. DO NOT configure any devices on channel 1 or on channel 15 on the HP-IB Interface Module. The Series III HP-IB Interface needs those channel memory areas for its own internal operations.

Thus, only those Series III computers with an attached HP-IB Interface are affected by this limitation. For information regarding further changes to startup procedures to follow when bringing up the system from a device on the HP-IB Interface, please refer to your updated System Manager/System Supervisor manual.

Using Labeled Tapes

by Mike Stodghill, Computer Systems Division

Labeled Tapes: Basic concepts

Labeled tapes are intended to provide:

1. A permanent identification for tape reels, or volumes;

2. Files which extend over more than one volume;

3. More than one file on a volume;

4. Retrieval of files by file name;

5. Additional security, to protect against invalid erasure or access to files.

Each tape volume, when first written, is assigned a unique identifier, up to six alphanumeric characters, to permanently identify it. This identifier is the Volume Name. It is often strictly numeric, and volumes in an installation's library can be stored and sorted by this number. A collection of volumes containing one or a related group of files is called a volume set.

Facilities affected are as follows:

- Command Changes--
  :FILE COMMAND
  :SHOWDEV COMMAND

- Intrinsic Changes
  FOPEN
  FREAD
  FWRITE
  FCONTROL
  FWRITELABEL
  FREADLABEL
  FCHECK

- New Console Messages

The user interfaces mentioned above are documented more fully in other H/P manuals. A working knowledge of these features is assumed. In the following discussion only those portions of the user interface which are relevant to MPE-TAPE-LABELS will be explained. For further information please see:

MPE COMMANDS REFERENCE MANUAL,
MPE INTRINSICS REFERENCE MANUAL,
MPE CONSOLE OPERATORS GUIDE.

:FILE COMMAND

MPE-TAPE-LABELS interfaces naturally with MPE. Users must specify optional file label information in the :FILE Command. The format of the file command is as follows:

```
:FILE formaldesignator=filename[[/lockword][.group]];

                                        F
     REC=[recsize][, blockfactor] ,     U   ,BINARY      ;CCTL
                                        V   ,ASCII       ;NOCCTL


     ;NOBUF              [;DEV=device][;CODE=filecode]
     ;BUF[=numbuffers]


     ;NOLABEL
     ;LABEL [=[volid],[type],[expirationdate],[seq]
```

filename - Up to eight alphanumeric characters beginning with an alphabetic character that names the file to be processed. This differs from standard MPE filereferences because there is no account associated with the file on tape and therefore the filereference parameter need not contain this information.

lockword - Up to eight alphanumeric characters beginning with an alphabetic character that the creator supplies in order to protect his file from illegal access. (optional parameter) If byte 54 of HDR1 label is a space there is no security protection for the file. If byte 54 is some other character then some kind of security is associated with the file. MPE tape labels will ignore security for IBM Standard labels. Security will be ignored for ANSI Standard labels unless byte 54=%230. If a lockword is specified on output then byte 54 of HDR1 label will be coded %230 and the lockword will be written in the portion of HDR2 label reserved for the operating system. This will enable HP3000 users to read ANSI Standard and IBM Standard tapes written by other systems, but will provide lockword security for ANSI Standard labeled tapes written by MPE 3000.

label - Specifies that this is a labeled tape equivalent to FOPTION bit 6 in FOPEN.

nolabel - Specifies that this is not a labeled tape - turns off FOPTION bit 6.

volid - Up to six printing characters that identify the volume. (Optional when opened for output.) See the article "File System Enhancements for the "D" Installation Tape" on page 8 for further details on specifying volid.

type - Three characters that specify label type. Options are:
ANS - ANSI standard labels
IBM - IBM standard labels
(optional parameter; default is FOPEN specified)

expiration date - Month/Day/Year. This is the expiration date of the file, or the date after which information contained in the file is no longer useful. The file can be overwritten after this date. (Optional parameter with default 00/00/00 meaning the file can be overwritten immediately.)

seq - Up to four characters that denote the position of the file in relation to other files on the tape. 0 will cause a search of all volumes until file name is found. If seq="ADDF" then the tape will be positioned to add a new file on the end of the volume (or last volume in a multi volume set). If seq="NEXT" then tape will be positioned at next file on the tape. If this is the first open then "NEXT" will make the position at the first file on the tape. If rewind file took place on the previous close then position will remain at the beginning of the previous file. (Optional parameter; default is FOPEN specified.)

```
:SHOWDEV COMMAND

:SHOWDEV TAPE


LDEV   AVAIL          OWNERSHIP         VOLID

  7    AVAIL
  8    UNAVAIL       #S81: 1 FILES     SADSAM       (ANSI)
  9    AVAIL
 12    AVAIL                                        (NOLABEL)
 13    AVAIL                           TAPE01       (ANSI)
 14    AVAIL                           666666       (IBM)
 15    AVAIL


:SHOWDEV 8

LDEV   AVAIL          OWNERSHIP         VOLID

  8    UNAVAIL       #S81: 1 FILES     SADSAM       (ANSI)

:SHOWDEV 7

LDEV   AVAIL          OWNERSHIP         VOLID

7      AVAIL
```

In the examples above logical devices 7, 8, 9, 12, 13, 14 & 15 are tape drives. Volumes are mounted on 8, 12, 13, and 14. Only LDEV 8 has an active volume. No volume is mounted on 7, 9, and 15. The volume on LDEV 12 is unlabeled. (Note that the DEN and ASSOCIATION fields have been omitted from the command output.)

INTRINSICS

FOPEN

The FOPEN intrinsic can also be used to specify label information. The format using this intrinsic is as follows:

```
INTEGER PROCEDURE FOPEN (formaldesignator, foptions, aoptions,
                         recsize, device, formsmessage,
                         userlables, blockfactor, numbuffers,
                         fillsize, numextents, initialloc,
                         filecode);
```

Users desiring to specify labeled tape information in the FOPEN intrinsic must set foptions parameter bit "6" to 1.

Additionally, tape label information must be placed in the forms-message parameter of the FOPEN intrinsic. (Note: Limit of 49 characters does not apply to labeled tapes.) The tape label information must be formatted as follows:

```
.[volid],[type],[expirationdate],[seq];
```

volumeid - Up to six printing characters that identify the volume. (Optional) Must begin with a period to distinguish the parameter as specifying tape label information.

type - Three alphabetic characters that denote label type information. Options are:

ANS - ANSI standard labels (default)
IBM - IBM standard labels

expiration date - Month/Day/Year. This is the expiration date of the file or the date after which information contained in the file is no longer useful. The file can be overwritten after this date. (Optional parameter with a default of 00/00/00 meaning the file can be overwritten immediately.) In a volume set, file expiration dates must be always equal to or earlier than the expiration date on the previous file. (See the article "File System Enhancements for the "D" Installation Tape" on page 8 for an alternate method of specifying expiration date.)

seq - Up to four characters that denote the position of the file in relation to other files on the tape. (Optional parameter with a default of "NEXT".) "0" will cause a search of all volumes until file name is found. If seq= "ADDF" then the tape will be positioned to add a new file on the end of the volume (or last volume in a multi volume set). If seq=NEXT then tape will be positioned at next file on the tape.

OPENING FILES ON LABELED TAPE

1. Must specify tape label information with a :FILE command or the formsmsg parameter of FOPEN intrinsic.

2. Must be an FOPEN/FCLOSE sequence for each file on a volume set. Simultaneous opens of files on labeled tapes are not allowed.

    EXAMPLE:

    To open file named INPUT on tape labeled TAPE01, use :FILE or FOPEN as follows:

    ```
    :FILE INDATA=INPUT;DEV=TAPE;LABEL=TAPE01,ANS,,0
    ```

    or for FOPEN, use:

    ```
    MOVE LABELINFO:=".TAPE01,ANS,,0;";

    FILENUM:=FOPEN(FILID,%1005,5,,DEV,LABELINFO);
    ```

    The file is the third file on the labeled tape.

VOLUME SET

Every tape that conforms to ANSI or IBM standards has a 6 character volumeid. The volumeid of the first volume is written in the file header labels of all files on the first volume and on any subsequent volumes. For reference purposes in this document the volumeid of the first volume will be referred to as the "volume set id" and the group of volumes with the same volume set id in their file header labels as the first volume will be referred to as a "volume set".

FREAD

READING FILES

The FREAD intrinsic is used to read files.

When a user's data file spans more than one tape volume, MPE-TAPE-LABELS will insure that the second volume of the set is mounted by issuing a mount message to the operator's console. The user's program will be suspended until the operator has completed mounting the volume. At that time the Automatic Volume Recognition function will wake the user program and processing will be continued.

CCG will be returned at actual EOF. CCG with a transmission log of 0 will be returned on attempts to read past EOF. CCG will not be returned for EOT where an actual file spills over onto another volume. After CCG on EOF, only FREADLABEL or FCLOSE will be accepted.

CCL and CCE remain as defined in the MPE Intrinsics Reference Manual.

FWRITE

WRITING FILES

The FWRITE intrinsic is used to write files.

When writing files on a scratch volume, or when subsequent volumes of a multivolume file are required, it is not necessary to specify volume identification in the :FILE command or the FOPEN intrinsic. MPE-TAPE-LABELS will request this information at the operator's console if it is not supplied.

Once the volume has been positioned by the FOPEN intrinsic, MPE-TAPE-LABELS expects to find either a tape mark, or a file header label belonging to an existing file.

If an existing header label is found, MPE-TAPE-LABELS will check the expiration date of the file. If the file has expired, MPE-TAPE-LABELS will overwrite the existing header label with a header label for the file to be written. If the file has not expired, MPE-TAPE-LABELS will verify through the operator's console that the file can be overwritten.

It is recommended that only the last file on a volume be overwritten, since files following the overwritten file will not be accessable at a later time. Files can be appended to existing volumes by specifying ADDF in the seq parameter of the :FILE command or the FOPEN intrinsic; this will cause the volume to be positioned immediately following the last file on the volume.

No CCL will be returned at EOT. If an FWRITE encounters EOT, MPE-TAPE-LABELS will write EOV1 and EOV2 labels and initiate reel switching. Otherwise condition code is as defined in the MPE Intrinsics Reference Manual.

WRITING A LABELED TAPE

Use :FILE command:

```
:FILE NEWTAPE1;DEV=TAPE;LABEL=FILO99,ANS,12/31/77,NEXT
```

or formsmsg parameter of FOPEN intrinsic:

```
BYTE ARRAY LABELID(0:79):=".FILO99,ANS,12/31/77,NEXT";

FNO2:=FOPEN(FILID2,%1004,5,,DEV,LABELID);
```

FCLOSE

CLOSING FILES

In order to maintain position when creating or reading a volume set, the disposition at FCLOSE should be either 2 (rewind to beginning of current file), or 3 (no rewind position at next file).

A disposition of 0 or 1 (rewind & unload) implies close of an entire volume set.

TAPE DISPOSITION

Users desiring to prevent rewinding of a tape file after it is closed or wishing to read data previously written can use the disposition parameter of the FCLOSE intrinsic.

Example:

A user desiring to close the current labeled file and open the same file again must close the file with a disposition code of 2. The file will be rewound to its beginning but will not be unloaded. Subsequent requests to open the file will not reposition the tape, if seq=NEXT or default.

FWRITELABEL/FREADLABEL

USER LABELS

MPE-TAPE-LABELS will allow users to read and write user labels to tape. User labels contain information specified by the programmer that is related to a particular file.

The intrinsics FREADLABEL and FWRITELABEL are used to create and read both header and trailer user labels.

In order to write a user header label, the programmer must issue the FWRITELABEL intrinsic prior to the execution of the first FWRITE intrinsic. After issuing the FWRITE intrinsic, MPE-TAPE-LABELS will terminate the data area and write user trailer labels for any FWRITELABEL request.

To read user header labels, the user must issue the FREADLABEL intrinsic prior to the execution of the first FREAD. The execution of the first FREAD causes MPE-TAPE-LABELS to skip past any unread user header labels.

User labels that follow file information are called user trailer labels. These labels must be written after the last FWRITE is issued to the file and before closing the file. After an FWRITELABEL is issued, MPE-TAPE-LABELS will not accept additional FWRITE's to the file. An error will be returned if an attempt is made. An attempt to read more user header or user trailer labels than actually exist will result in CCG.

User trailer labels can be read by issuing an FREADLABEL after any of the data area of the file is read or processed. It is assumed by MPE-TAPE-LABELS that the next action to be taken against a file after reading user trailer labels will be to close or rewind the file.

*NOTE: All ANSI Standard and IBM Standard labels are 80 bytes in length. User header labels and user trailer labels should conform to this standard. Any attempt to write or read other than 80 bytes (or 40 words) will result in CCL.

FCONTROL

There are certain functions of the FCONTROL intrinsic that cannot be applied when processing labeled tapes. This is because labeled tapes are file oriented, with the possibility of multiple files on a single tape, rather than device or single file oriented as is the case with unlabeled tapes. Thus FCONTROL operations such as "Space forward to tape mark" cannot be allowed when using labeled tapes.

Certain other FCONTROL functions might yield unexpected results if the user issued them. An example here is that the "Rewind file" control will position the tape to the beginning of the opened file, not necessarily the beginning of the volume. A list of the FCONTROL control codes and their new or modified meanings follows:

Control code	Meaning	Operating With Labeled Tapes
5	Rewind File *	Position tape at the beginning of the opened file on the volume. (Start of user Header Labels if any.)
6	Write End-Of-File	Not allowed for labeled tapes.
7	Space Forward to Tape Mark	Space forward to start of User Trailer Labels, if any.
8	Space Backward to * Tape Mark	Rewind file to start of User Header Labels, if any.
9	Rewind and Unload	Not allowed for labeled tapes.

*In the case of a multi-volume configuration, when the user requests rewinding of the file that spans the volume, MPE-TAPE-LABELS will not request that the previous volume be mounted. MPE-TAPE-LABELS will only rewind the file to the beginning of the correct volume.

FCHECK INTRINSIC

TAPE LABEL ASSOCIATED ERRORS:

116 INVALID TAPE LABEL FOPEN PARAMETERS
117 ATTEMPT TO WRITE ON UNEXPIRED TAPE FILE
118 INVALID HEADER OR TRAILER TAPE LABEL
119 I/O ERROR POSITIONING TAPE FOR TAPE LABELS
120 ATTEMPT TO WRITE IBM-STANDARD TAPE LABEL
121 TAPE LABEL LOCKWORD VIOLATION
122 TAPE LABEL TABLE OVERFLOW
123 END OF TAPE VOLUME SET
124 ATTEMPT TO APPEND LABELED TAPE

REEL SWITCHING

When processing a Multivolume data set, the user should specify the first volume to be processed in the :FILE command or the FOPEN intrinsic. If omitted, as one might if a scratch tape is wanted, the operator will be solicited for the volume name. When writing and an end-of-tape reflective marker is encountered or when reading and an EOV label is encountered, MPE will request that the next volume be mounted on the same LDEV. If writing, a labeled tape mounted to be rewritten will be checked for expiration.

TAPE-SECURITY

MPE provides two kinds of tape file protection: Protection against accidental destruction of tape file information, and protection of private information on tape files.

Protection against accidental destruction of tape files is inherent in the operation of MPE-TAPE-LABELS because the user must request specific volume and file information before a file can be overwritten. Protection of private information is provided through the use of the lockword parameter of the :FILE command and FOPEN intrinsic. When a file is created with a lockword specified, all subsequent users of the file must provide that lockword. Thus the lockword parameter serves two purposes: First to initially establish a lockword for a file and then to provide a means of supplying the lockword to subsequent users of the file.

CONSOLE MESSAGES

1. hr:min/#jsnum/pin/Vol(unlabeled) mounted on LDEV#n

2. hr:min/#jsnum/pin/Vol(ANSI) mounted on LDEV#n

3. hr:min/#jsnum/pin/Vol(IBM) mounted on LDEV#n

One of the above messages (depending on whether the tape is unlabeled, has an ANSI-standard label, or has an IBM-standard

label) is displayed when a tape is recognized. Note that #jsnum is not displayed unless the mount was in response to a mount request (otherwise the jsnum would not be known).

4. hr:min/#jsnum/pin/Mount tape of volumset volid

This message is displayed when a labeled tape is opened but has not been mounted. A reply of 0 will abort the FOPEN, otherwise the operator should mount the requested volume without any reply, or mount a blank tape and reply with the LDEV #.

5. hr:min/#jsnum/pin/Mount next volume of set volid on LDEV#n

6. hr:min/#jsnum/pin/Mount prior volume of set volid on LDEV#n

This message is displayed when a file spans more than one volume and it is necessary to switch volumes. No reply is necessary, the operator merely mounts the appropriate volume on the specified device.

7. hr:min/#jsnum/pin/Vol ID for volume of set volid on LDEV#n

This message is displayed when an unlabeled tape is mounted in response to message 5. The operator is expected to supply volume header identification which will be written on the tape by MPE.

8. hr:min/#jsnum/pin/VOLUME ID FOR filename (MAX CHAR.=6)?

This message is displayed when a labeled tape is opened and no volume header information is supplied. The operator is expected to reply with volume header identification.

9. hr:min/#jsnum/pin/OK to write on unexpired volume on LDEV#n? (Y/N)

This message is displayed when a labeled unexpired tape is opened as unlabeled, or when a labeled unexpired tape is mounted in response to message 4 or 5.

COBOL vs COBOL II
Use of 'Perform' and 'Goto' Statements

By John Pavone, Information Networks Division

This article presents some differences in the execution of PERFORM statements between the two compilers and more specifically it illustrates some potential problems which can occur to users who convert executable COBOL programs to COBOLII.

The COBOL/3000 compiler utilizes a run-time table for controlling exit returns from PERFORM statements. User's interested in the run-time table environment are referred to the Communicator issue #25 article, by Tony Lemberger, entitled "COBOLII/3000-COBOL/3000 Stack Layout Comparisons".

The COBOLII compiler utilizes firmware for controlling exit returns from PERFORM statements. Users interested in how this is implemented are referred to the Communicator issue #26 article, by Greg Gloss, entitled "Debugging COBOLII/3000 Programs".

The implementation method employed by COBOLII for controlling the exits enforces a more logical execution of PERFORM functions and also allows the user to easily incorporate a level of programming style, commonly referred to as "structured programming", within the program design.

A number of important differences can result at run time, depending on the techniques, used in COBOL, when the program is converted to COBOLII. In some cases erratic program flow or unexpected program aborts may occur, which will require additional conversion time for program logic analysis and/or debug to identify and locate. These differences may be defined as follows:

1. COBOLII allows recursiveness in cases of "common" exit points, e.g. The statement PERFORM B THRU C within a PERFORM A THRU C.

   COBOLII results in two exits when going through C, whereas COBOL results in only one exit to the latest PERFORM reference (B thru C) and a fall-through for (A thru C).

2. PERFORM exits under COBOLII, when nested, should occur in the reverse order as PERFORM starts, otherwise fall-throughs occur instead of exits. On the other hand in COBOL exits may occur in any order. The fall-throughs, in COBOLII, will happen when GO TO's are used to prematurely exit out of PERFORM ranges.

3. If a non-dynamic subprogram is exited while a PERFORM exit is still active, under COBOL, the next subprogram CALL may cause the exit to be executed without having a PERFORM active. In COBOLII, this will not happen.

4. In COBOLII, GO TO and PERFORM statements may appear within the range of the PERFORM, provided that the logical paths developed by these statements eventually lead back to the last statement of the last paragraph of the range. In COBOL exiting out of the range via a GO TO statement, in some cases, may not affect the intended execution of the program. In these cases the user has been allowed to violate a rule, without penalty, until a conversion to COBOLII is done.

The following examples are presented to illustrate the rules presented above with specific emphasis placed on those cases where execution of a COBOL/3000 and COBOLII/3000 program would not be the same.

Example 1: This example illustrates the enhanced capability of COBOLII which allows the use of common exit points for nested performs.

```
PROCEDURE DIVISION.  DO-IT.
     PERFORM A THRU D.  STOP RUN.
A.
     PERFORM B THRU D.
B.
     .....
C.
     .....
D.
     PERFORM F THRU G.
E.
     .....
F.
     .....
G.
     .....
```

Note that the common exit point is paragraph "D". COBOLII will allow this case to execute while COBOL will cause a fall-thru following the second exit at "D" into paragraphs E,F & G. The trace paths would execute as follows:

COBOLII- DO-IT, A, B, C, D, F, G, B, C, D, F, G, STOP RUN.

COBOL - DO-IT, A, B, C, D, F, G, B, C, D, F, G, E, F, G, ABORT.

Example 2: This example illustrates the enhanced capability of COBOLII which allows the inner perform of a nested perform to pass through the exit of the outer perform.

```
PROCEDURE DIVISION.  DO-IT.
     PERFORM A THRU C.  STOP RUN.
A.
     PERFORM B THRU D.
B.
     .....
```

```
C.
      .....
D.
      .....
E.
      .....
```

Note that the exit point C will return control to the outer loop, under COBOL, whereas it will pass thru successfully under COBOLII.

The trace paths would execute, as follows:

COBOLII- DO-IT, A, B, C, D, B, C, STOP RUN.

COBOL - DO-IT, A, B, C, STOP RUN.

Example 3: This example illustrates a violation of a perform rule which will execute successfully under COBOL, but not COBOLII. The violation involves the use of GO TO statements which cause the program flow to leave the range of an active perform via the GO TO instead of the defined perform exit.

```
PROCEDURE DIVISION.  DO-IT.

      PERFORM A THRU E.  STOP RUN.
A.
      PERFORM F THRU G.
B.
      PERFORM C THRU D.
C.
      .....
D.
      .....
E.
      .....
F.
      GO TO B.
G.
      .....
```

Note that the GO TO B, in this case, causes a loop to occur under COBOLII, but not COBOL. In other cases the run-time results could be unpredictable.

The trace paths would execute as follows:

COBOLII: DO-IT, A, F, (B, C, D, C, D, E, F), (B, C, D, C, D, E, F), (B, C, D, C, D, E, F) .....ETC.

COBOL : DO-IT, A, F, B, C, D, C, D, E, STOP RUN.

Example 4: This example illustrates a correction to example 3, which will allow correct program logic while still permitting the use of GO TO statements within active performs. In addition, it also illustrates the use of GO TO's to exit out and re-enter the range of the active perform, which works under both COBOL and COBOLII.

```
PROCEDURE DIVISION.  DO-IT.

      PERFORM A THRU E.  STOP RUN.
A.
      PERFORM F THRU G-XIT.
B.
      PERFORM C THRU D.
C.
      GO TO H.
D.
      .....
E.
      .....
F.
      GO TO G-XIT.
G.
      .....
G-XIT.
      EXIT.
H.
      GO TO D.
```

Note the addition of a defined exit for G and the exit/return GO TO's, maintaining program logic flow for the PERFORM C THRU D.

The trace path would execute, as follows, for both compilers:

DO-IT, A, F, B, C, H, D, C, H, D, E, STOP RUN.

CONCLUSION

As you may have already concluded, following your review of the above illustrations, it is very possible that the results of COBOL programs, which executed CORRECTLY?? may not necessarily execute the same when converted to COBOLII. This article is not intended to imply that upwards compatibility from COBOL to COBOLII does not exist, because it does in fact exist, given that programs developed under COBOL were written according to the ANSI STANDARD rules for PERFORM statements. This article is intended to highlight the fact that where the rules have been violated under COBOL, the program flow may have produced the desired results because of how the PERFORM RETURN TABLE was implemented. The change in design of PERFORM under COBOLII, via the run-time stack marker, no longer allows the violations to execute the same.

RECOMMENDATION

In order to develop programs, which will execute on any computer system and be within the ANSI STANDARD for PERFORM statement operation, the following two simple rules are presented:

1. DO NOT ALLOW nested performs to exit at a COMMON EXIT POINT.

2. DO ensure that, when performs are nested, the inner perform is allowed to reach its exit point before reaching any exit of an outer perform.

COBOL TO COBOLII CONVERSION PROBLEMS

When a program, which executes successfully under COBOL, is converted to COBOLII and does not execute or executes with different results a prime suspect of the problem cause is a nested perform which does not meet the simple rules presented above. If this is the case, the symptom will ususally be associated with a perform that does not exit correctly and therefore causes execution to fall-through the code. Location of these problems will require examination of the coding techniques utilized in developing the program and perhaps some debug operations to check-out the flow of any suspected operations.

Some of the key elements to look for are:

- GO TO's, within the range of performs, that transfer control outside the range instead of to the exit when execution of the remaining code in the range is not desired. Note: If the GO TO is prematurely exiting from a 'PERFORM ... n TIMES' the statement may have to be changed to a 'PERFORM ... UNTIL condition'.

- Inner performs which cross over the exits of outer performs.

- Performs, using a COMMON EXIT, that may have been executing and producing the desired result.

Calling SPL from Other Languages on the HP 3000

By Mark Cousins, NSR Palo Alto

There are a number of things to consider when writing SPL procedures that are to be called from other languages. Not all languages pass parameters in the same way and some have restrictions as to their ability to call function procedures, OPTION VARIABLE, and so forth. This note summarizes these restrictions for BASIC, COBOL, COBOL II, and FORTRAN.

First let's define a few terms. There are two ways to pass a parameter to a procedure: by REFERENCE and by VALUE. Passing a parameter by reference means that the 16-bit ADDRESS of the variable is passed on the stack; the called procedure refers to this parameter via indirect memory reference instructions (LOAD Q-n,I and STOR Q-n, I). Passing a parameter by value means that the actual contents of the variable (1, 2, or 4 words) are passed on the stack; the called procedure refers to this parameter via direct memory reference instructions (LOAD Q-n and STOR Q-n). One of the consequences of all of this is that if the called procedure modifies a call-by-reference parameter, the caller's variable is modified; for call-by-value parameters, only the "temporary" copy in Q-minus storage is changed (the caller's version retains its old value).

OPTION VARIABLE is a facility that provides the ability to call a procedure with a varying number of parameters. The called procedure will expect a "bit mask" in Q-4 (and Q-5 if there are more than 16 parameters) with bits set indicating which parameters are present. Parameters are always passed in the same Q-minus addresses; the Q-minus locations for parameters which are omitted have undefined values. It is up to the called procedure to examine the bit mask and to not access parameters which are not passed on any particular call.

A function procedure is one which returns a value in place of its name; it therefore can be called from an expression and the value that it returns will be used in the expression. This value is stored in the stack just before (lower address) the parameters to the procedure. It is the responsibility of the caller to dispose of or use the return value properly. An example of such a procedure is the BINARY intrinsic.

Because the various languages have differing capabilities for dealing with the various aspects of procedure calls, the SPL coder needs to be aware of what each language does. Below are summarized the things that need to be considered for each language.

COBOL

- All parameters are passed as WORD addresses (call-by-reference). There is one exception: you can pass

the MPE file number for a file opened with the OPEN verb by passing the FD-name to a procedure; this is passed as a 16-bit integer by value.

- COBOL has no way of coping with the return value of a function procedure; an extra value will be left on the stack which can and probably will disrupt program execution. Don't call function procedures from COBOL.

- There is no way for COBOL to generate the bit mask required by OPTION VARIABLE procedures, so these cannot be called either. Since it is impossible to pass a parameter from COBOL by value, you can't generate the bit mask yourself.

- Here is how the COBOL data types map to SPL data types:

COMPUTATIONAL	
1-4 digits	INTEGER
5-9 digits	DOUBLE
COMPUTATIONAL-3	SPL has no PACKED DECIMAL capability; you must access this as a byte array and generate the machine instructions yourself. Note that COBOL passes a WORD address for this; you will need to use an equivalenced byte array.
DISPLAY	Passed as LOGICAL (array). You will usually want to equivalence a byte array to the passed parameter and access the data this way.

Note that COBOL has no equivalent of REAL or LONG.

FORTRAN

- FORTRAN passes all parameters by reference unless the parameter is enclosed in backslashes, in which case it is passed by value. You may use a constant or expression in a call; if it is not enclosed in backslashes, a temporary cell is created and the address of the cell is passed.

- FORTRAN may call function procedures normally (external function).

- If you are calling an OPTION VARIABLE procedure, you must calculate the bit mask required and pass it as a constant by value as the LAST (or last two) parameter(s). See below for form of the bit mask.

- Here is how FORTRAN data types map to SPL data types:

INTEGER/INTEGER*2	INTEGER
INTEGER*4	DOUBLE
REAL	REAL
DOUBLE PRECISION	LONG
CHARACTER*n	BYTE ARRAY

- Should you want to call an intrinsic, your life will be greatly simplified if you name the intrinsic in a SYSTEM INTRINSIC statement. Then FORTRAN will take care of the OPTION VARIABLE mask, passing of parameters by reference or value, and so on.

BASIC

- BASIC passes all parameters by reference. There is no way to override this; if you pass a constant or expression, a temporary cell is created and the address of the cell is passed.

- BASIC, like COBOL, can't handle the return value from a function procedure. Likewise, it has no ability to generate an OPTION VARIABLE bit mask. Because all parameters are call-by-reference, you cannot generate a proper bit mask either.

- BASIC passes a parameter type descriptor just in front of (lower memory address) the first parameter. The called procedure may use this or ignore it - see the BASIC Interpreter reference manual for details. This descriptor does not interfere with the normal addresses of the parameters.

- Here is how BASIC data types map to SPL data types:

REAL/undeclared	REAL
LONG	LONG
INTEGER	INTEGER
String (x$)	BYTE ARRAY

  Please keep in mind that the default constant in BASIC is type-REAL. To pass an integer, you must either store the value into an integer variable and pass the variable or use the following construct:

```
DEF INTEGER FNI(N)=N
     .  .  .
CALL proc(FNI(4))
```

This will pass the 4 as an integer instead of a real number.

Arrays and strings have physical and logical length information stored in the -2 and -1 elements of the array. (See the reference manual.) The point to note here is that if you change the length of a string or array, you must update the logical length so that BASIC knows what you did. Two-dimensional arrays and string arrays have length information at the beginning of each major dimension or string element.

(See below for a discussion on converting byte addresses to word addresses.)

COBOL II

- Much like FORTRAN, COBOL II passes all parameters by reference unless the parameter is enclosed in backslashes, in which case it is passed by value.

- All parameters are passed as WORD addresses unless an @ is used in front of the parameter name, in which case a BYTE address is passed.

- If you are calling a function procedure, an extension to the CALL statement (the GIVING clause, as in CALL proc USING parm GIVING value) allows you to pick up the return value; you MUST use this construct if you are calling a function procedure (even if you have no use for the return value) so that the stack is decremented properly.

- As with FORTRAN, you can generate the bit mask for OPTION VARIABLE procedures by passing it by value as the last parameter(s).

- COBOL II allows you to call intrinsics via the CALL INTRINSIC statement, relieving you of worrying about value v. reference, byte addressing, the OPTION VARIABLE mask, and so forth.

- The data types are precisely the same as for COBOL, above.

OPTION VARIABLE mask

The OPTION VARIABLE mask is one word at Q-4 (or two words at Q-5 and Q-4 if there are more than 16 parameters) that describes which parameters are present. The RIGHTMOST bit (bit 15 in HP3000 nomenclature) corresponds to the rightmost (last) parameter; bit 14 refers to the next-to-last, and so forth on back to the first parameter. A 1 bit means the parameter is present; 0

means that the parameter was omitted and it should not be accessed.

For example, suppose we have the following procedure head:

```
PROCEDURE upshift(string,length,result);
VALUE length;
BYTE ARRAY string;
INTEGER length,result;
OPTION VARIABLE;
```

and we wish to call this from FORTRAN. What would be the proper CALL statement? Since there are three parameters, the last three bits of the mask would be used. If all parameters were included, the call would look like this:

```
CALL UPSHIFT(CHARSTRING,\LEN\,IRESULT,\%7L\)
```

If, say, the last parameter (RESULT) were omitted, the call would be:

```
CALL UPSHIFT(CHARSTRING,\LEN\,\0\,\%6L\)
```

The zero as the third parameter is required as a place holder.

Byte to word address conversion

It is sometimes desirable (or necessary) to convert a passed byte address to a word address (so that the array can be passed to the file system intrinsics, for example). You will find that if you attempt to equivalence a word array back to a passed byte array you will get a warning "ARITHMETIC RIGHT SHIFT EMITTED." What this is saying is that the SPL compiler is emitting an ASR 1 instruction to convert the byte address to a word address, and you are being warned because this is not always the correct thing to do. The reason for this is that it is possible to have byte addresses that point to the DB-minus area (in fact, BASIC does this all the time) but it is impossible to tell if an address is in the DB-minus area or is simply a very large DB-plus byte address without looking at the registers. Here is a foolproof procedure that will generate the proper word address given any byte address:

```
LOGICAL PROCEDURE wordadr(byteadr);
VALUE byteadr;
BYTE POINTER byteadr;
BEGIN
  tos:=tos:=@byteadr & LSR(1);  << Logical divide by 2 >>
  PUSH(z);  << Get Z-register >>
  IF tos>tos then tos.(0:1):=1; <<If in DB-minus, fix sign>>
  wordadr:=tos
END;  << wordadr >>
```

Sample call:

```
     PROCEDURE sample(string);
     BYTE ARRAY string;
     BEGIN
       POINTER stringp;  << Word pointer >>
-->    @stringp:=wordadr(string);
          .  .  .
```

Editor's Note: The following article is a reprint of an earlier press release.

Correspondence-Quality Printer for Hewlett-Packard Computer Systems

A new daisywheel printer joins the list of peripheral products now available for Hewlett-Packard computer systems. The HP 2601A produces letter-quality output, suitable for word-processing, text and document applications. It is especially recommended for use with HP's recently-introduced Text and Document Processing (TDP/3000) software system for these purposes.

Enhancements that usually require additional applications software are built into the printer and are fully supported by TDP/3000. These include proportional spacing, automatic underlining, automatic centering, and right justification, all built into the printer's firmware. Bold and shadow print may be used as highlighting tools to accentuate key phrases in reports produced.

The HP 2601A operates either with a metal wheel, at 32 characters per second, or with a low-cost plastic wheel at 40 cps. Both are available in a variety of languages and type styles. Other features are absolute tabs, margin controls, bi-directional printing, and reverse paper feed.

Standard with the HP 2601A is a friction feed platen for use both with single-sheet and continuous forms. HP also offers bidirectional forms tractors for more precise control over feeding of continuous forms.

The HP 2601A is supplied with an RS-232-C interface. It operates with all HP 3000 and HP 1000 Computer Systems, and with HP terminals, including the HP 2645, 2647, 2642, 2626, and 2624. Support of specific features varies among these.

Documentation

The catalog of customer publications at the end of this section lists the currently available customer manuals for HP 3000 Computer Systems products. This list supersedes the catalogs in previous issues of the COMMUNICATOR.

Purchasing

Customers may purchase copies of new manuals, new editions and updates by either Direct Phone Order or by placing orders through their local HP Sales and Service Office.

The Direct Phone Order numbers are (800) 538-8787 (toll free) and, in California, (408) 738-4133 (collect). Calls should be made between 9:00 a.m. and 5:00 p.m. in the caller's time zone. Most orders will be shipped within 24 hours.

The addresses and telephone numbers of local HP Sales and Service Offices are listed in the back of all customer manuals.

Prices of HP documentation are subject to change without notice.

To obtain a manual update, the customer must purchase the manual to which it pertains. The latest edition of the manual, along with the update, will then be sent to the customer.

Computer Documentation Index

The Spring 1981 Computer Documentation Index available from Hewlett-Packard includes manuals, binders and a variety of non-promotional publications for HP computer systems (from the HP-85 through the HP 3000, and for terminals and peripherals, including plotters).

The Index is arranged "By Subject (model number)", which is useful when searching for documentation items where the specific part number is not known; and a "Numerical" listing which provides a fast way to find documentation when the part number is already known. Both listings include description, print date, latest update, and (U.S. Edition only) the current list price.

To receive a copy of this useful reference, please mail your request to:

Hewlett-Packard
Computer Supplies Operation
P.O. Box 60008
Sunnyvale, California 94008

Terms

A few words about documentation terms and procedures:

NEW — The first printing of the first edition. When first printed, a manual is assigned a part number that is retained for the life of the manual.

UPDATE — A supplement to an existing manual which contains new or changed information. Manual updates, which are issued between editions, contain additional or replacement pages to be merged into the manual by the customer.

Updates are generally issued at the same time Installation Tapes (ITs) are issued. However, THERE IS NO DIRECT CORRELATION BETWEEN SOFTWARE FIXES AND MANUAL UPDATES. Software enhancements that require documentation changes will be accompanied by manual updates, but software fixes and manual corrections may be made independently.

Updates are retroactively inclusive; that is, whenever successive updates are issued, the later update will contain the previous one(s). This means that you need obtain only the latest update to have all the information added or changed since the last printing of the manual.

Manual updates do not have part numbers. They are numbered sequentially from the time the last edition was issued.

NEW EDITION — A complete revision of a manual; obsoletes all previous editions of the manual and its updates.

A new edition is issued when, due to the scope of the changes involved, it is impractical to issue a manual update.

The date on the title page and back cover of every manual is the printing date of the current edition. This date changes only when a new edition is published. A list of the dates of the manual's previous editions and updates (if any) is kept on the Printing

History page of every manual. Publication of a new edition does not affect the part number of a manual.

If further updates are required, they are made to the new edition. The update numbers run sequentially, starting from the latest edition.

Console Operator's Guide
Part Number 32002-90004
May 1981

This new edition of the Console Operator's Guide applies to the entire HP3000 Series family of computer systems: the Series II, III, 30, 33, and 44.

Although there are some major physical differences between the systems, they all use essentially the same operating system. Thus, the commands, syntax, and general operating instructions are common to each of the above mentioned systems.

The first edition of the new Console Operator's Guide incorporates all of the material previously contained in three separate manuals, plus all new information pertinent to the 2131 ("D" IT) software release.

TDP/3000 Reference Manual
Part Number 36578-90001
April 1981

This publication is the reference manual for Text and Document Processor/3000, a command driven text processing system used to create, modify, and format text files for draft or final printing of a variety of documents. TDP/3000 can be used to prepare many different kinds of documents: manuals, multi-column tabular reports, form letters, and papers containing mathematical/ scientific expressions, to name a few. This manual is written for the user who is familiar with TDP/3000, EDIT/3000, or any text processing system. It is intended to be used as a reference guide, rather than as a self-study introduction to TDP.

Using TDP/3000
Part Number 36578-90002
November 1980

Using TDP/3000 provides a step-by-step introduction to text editing and document formatting using Text and Document Processor/3000. The manual is aimed at the inexperienced user who wishes to learn enough about TDP/3000 to create most common documents. The most often used editing and formatting commands can be learned through text and a series of exercises which the reader must complete at the terminal using TDP.

TDP/3000 Quick Reference Manual
Part Number 36578-90003
November 1980

The Quick Reference Manual contains a one-line description of the most often used TDP/3000 commands in an easy-to-use spiral binder. This manual was written to serve as a quick reference for the user who is new to computers and text processing.

NEW EDITIONS

DSN/MTS Multipoint Terminal Software Reference Manual
Part Number 32193-90002
August 1981 — Fourth Edition

This new edition describes the use of IBM 3270 devices on MTS lines, and introduces the support of two new products on MTS. The new products are the HP 13264A Data Link Adapter for HP 262X terminals, and the HP 3074M Modem Link Adapter for remote asynchronous communications. In this edition, MTS is referred to for the first time by its new name, DSN/MTS. DSN is the acronym for Distributed Systems Network, the designation for the HP Information Networks product family. Appendix D, "Configuring DSN/MTS into MPE" has been revised so that the step numbers conform to the MPE IV step numbering sequence.

DSN/MRJE Multileaving Remote Job Entry Reference Manual
Part Number 32192-90001
August 1981 — Fourth Edition

In the fourth edition, the manual has been changed to reflect the renaming of MRJE/3000 to DSN/MRJE. The manual has also been modified to document the use of the HP 2680A Laser Printer for printing MRJE output. The fourth edition includes an expanded discussion of the management of unsolicited output, documentation of the new MRJECONTROL command for checking software version, fix and update levels (:MRJECONTROL CHECK), enhanced error message definitions and descriptions of recovery procedures, and an MRJE installation procedure, located in Appendix B.

DSN/IMF Interactive Mainframe Facility Reference Manual
Part Number 32229-90001
September 1981 — Third Edition

The product previously called IML/3000 is being reidentified as Distributed Systems Network/Interactive Mainframe Facility (DSN/IMF), and the manual's title now reflects the new name. This new edition also documents the following additional changes. The usage of either Binary Synchronous Communications (BSC), or 3271 Synchronous Data Link Control (SDLC) as an SNA "physical unit one" is now allowed. IDF is now called the Pass Thru Facility, and the Command Interpreter of MPE is changed to accept the commands IMF, IMFCONTROL, and IMFMGR. A Data Stream mode with intrinsics is added to obtain untranslated data, and to allow the transfer of files between the host and HP3000 systems. Data Block restrictions have also been removed.

HP 30018A/30019A Add-On ADCC Update #1
Installation Manual
Part Number 30070-90023
June 1981

This update provides a revised illustration of the processor power switches to depict the new rotary switch now being used on the HP 3000 Series 30 Computer Systems.

DS/3000 Reference Manual Update #5
Part Number 32190-90001
May 1981

This update documents the Network File Transfer (NFT) facility. NFT is an enhancement to DS/3000 which provides the ability to transfer disc files between HP 3000 systems more efficiently than the FCOPY utility and remote file access. The NFT program, when initiated over a DS/3000 data communication link, can copy a file to or from any other HP 3000 computer which also has the NFT capability.

MPE System Utilities Update #7
Reference Manual
Part Number 30000-90044
March 1981

This update contains several error corrections and clarifications, including changes to the SPOOK utility to aid in the use of the 2680A page printer.

MPE Software Pocket Guide Update #1
Part Number 30000-90049
April 1981

This update includes changes and enhancements to MPE software made in the last product release.

MPE Commands Refernece Manual Update #1
Part number 30000-90009
April 1981

This is the first update to the fourth edition, and includes the addition of two parameters to the :FILE command. The environment (;ENV=) parameter is used in conjunction with the HP 2680A Laser Printing System, and the density (:DEN=) parameter with the HP 7976A Magnetic Tape Drive.

MPE IV System Manager/System Supervisor Reference Manual
Part Number 30000-90014
February 1981

Update #1

The first update to the fourth edition of the System Manager/ Supervisor Reference Manual is intended for HP 3000 Computer System users assigned the System Manager, Account Manager, or System Supervisor capabilities under the MPE IV operating system. The major highlights of this update edition are: the 7976A magnetic tape unit, the HP-IB Interface Module on the Series III, a revision of Appendix C, and the 2680A Laser Printing System.

MPE Intrinsics Reference Manual
Part Number 30000-90010
July 1981

Update #1

This first update to the third edition is highlighted by the addition of four user logging intrinsics, changes to the FCHECK, FFILEINFO and FOPEN intrinsics, and the addition of the FDEVICE-CONTROL intrinsic to accomodate the new 2680A Laser Page Printer. New information on enhancements to the Tape Labels Facility is also presented in this update.

KSAM/3000 Reference Manual
Part Number 30000-90079
May 1981

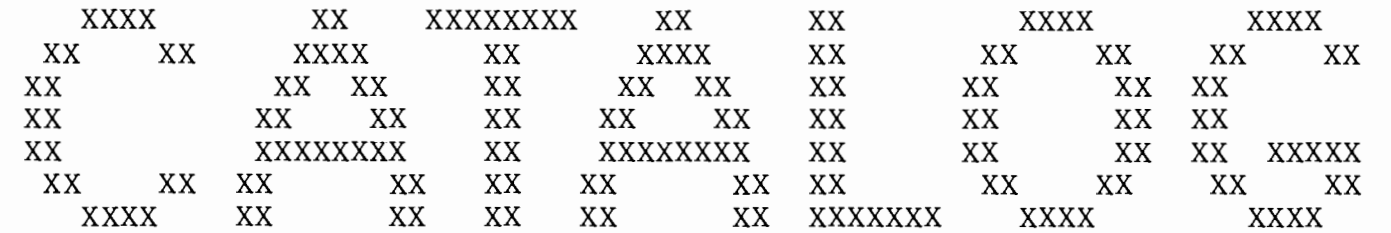
Update #1

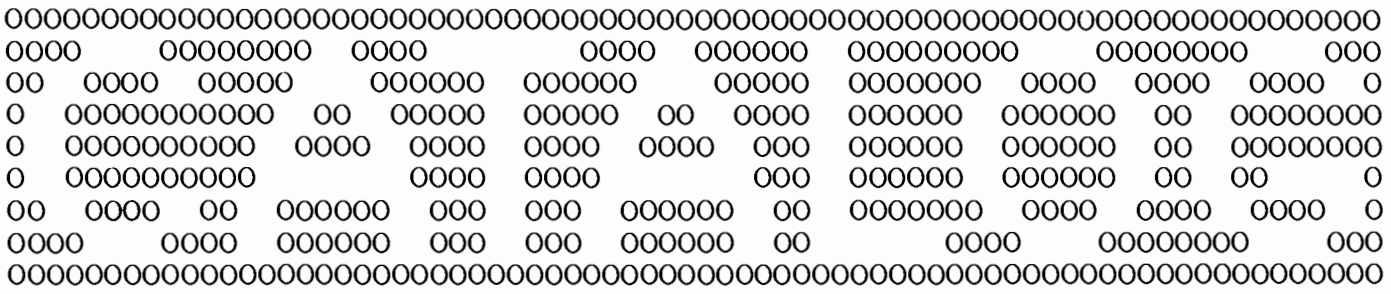
This update is a result of changes suggested by users inside and outside the company. In some cases, explanations within the text have been enhanced; in others, erroneous information has been corrected. These changes do not reflect any changes in the software.

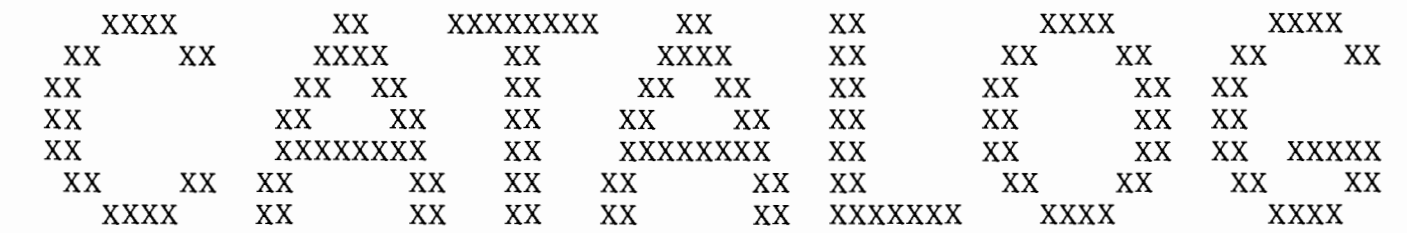
DS/3000 Reference Manual
Part Number 32190-90001
May 1981

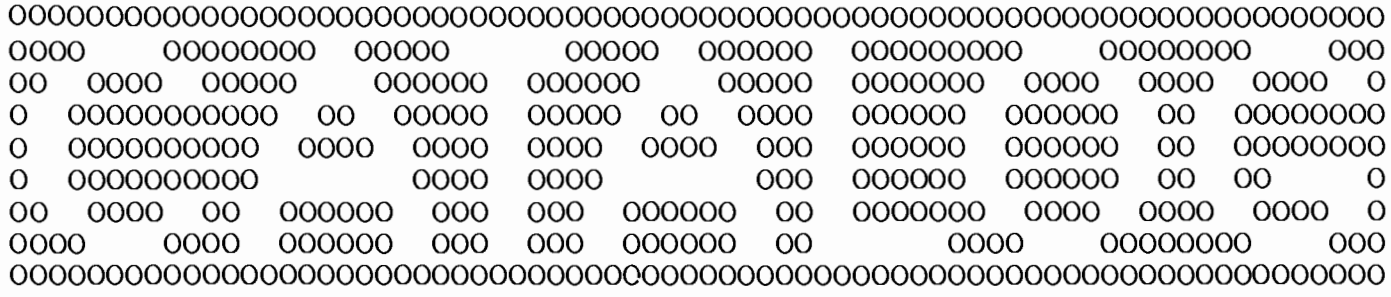
Update #5

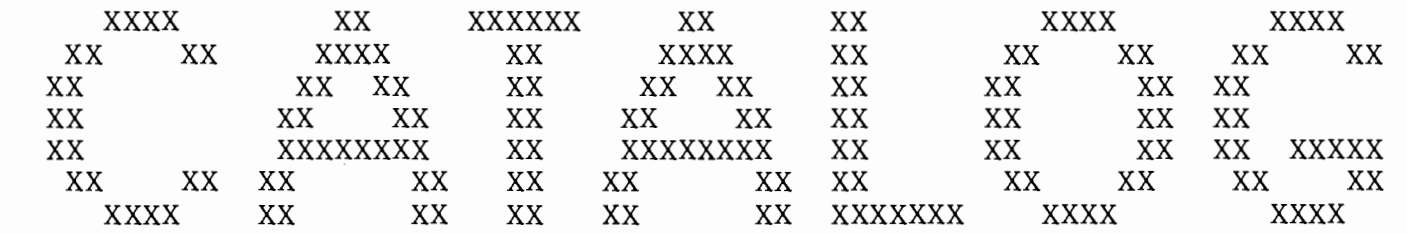
This update includes the Network File Transfer capability of DS/3000, which provides a means of efficiently transferring disc files between two HP 3000 Systems. The systems involved can be a local and a remote HP 3000, or two remote HP 3000s that are in session with a local HP 3000.











KEY

Manuals that are new or have changed since the last edition of this catalog are noted by an asterisk (*) in the leftmost column. An asterisk in the "Price" column indicates that the price of the manual was not available at the time the catalog was printed.

If the V (version) column contains a #, the manual is applicable to systems running MPE IV and to those running MPE C. Manuals which apply to MPE C systems only are listed under "MPE C MANUALS".

HP 3000 COMPUTER SYSTEMS

SYSTEM MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
Using the HP 3000: An Introduction to Interactive Programming	#	03000-90121	10.00	4/79	
*MPE Commands Reference Manual		30000-90009	16.75	1/81	4/81
*MPE Intrinsics Reference Manual		30000-90010	19.25	1/81	7/81
MPE Segmenter Reference Manual	#	30000-90011	3.50	2/77	
MPE Debug/Stack Dump Reference Manual	#	30000-90012	4.50	9/76	10/80
*Console Operator's Guide		32002-90004	@	5/81	

SYSTEM MANUALS (continued)

Manual Title	V	Part Number	Price	Print Date	Up-dated
*System Manager/System Supervisor Manual		30000-90014	17.25	2/81	5/81
Error Messages and Recovery Manual		30000-90015	18.50	###	
HP 3000 Computer System Machine Instruction Set		30000-90022	8.75	2/80	
*MPE System Utilities Reference Manual		30000-90044	4.50	3/77	3/81
Index to MPE Reference Documents		30000-90045	4.00	5/81	
*Software Pocket Guide		30000-90049	9.25	2/80	4/81
Using Files	#	30000-90102	15.50	4/78	

These manuals have been temporarily removed from circulation.

@ Price has not been established.

SUBSYSTEM MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
EDIT Reference Manual	#	03000-90012	7.25	8/80	
Trace Reference Manual	#	03000-90015	4.50	6/76	
FCOPY Reference Manual	#	03000-90064	4.75	7/80	
Scientific Library Reference Manual		30000-90027	6.75	6/76	2/77
Compiler Library Reference Manual		30000-90028	13.50	11/76	
FLEXIBLE DISCCOPY/3000		32199-90001	6.00	8/80	
SORT Reference Manual	#	32214-90001	8.50	3/80	

HP 3000 COMPUTER SYSTEMS

LANGUAGE MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
BASIC for Beginners	#	03000-90025	6.00	11/72	
BASIC/3000 Pocket Guide	#	03000-90050	1.25	9/74	
System Programming Language Reference Manual	#	30000-90024	12.00	9/76	2/77
System Programming Language Textbook	#	30000-90025	11.50	6/76	1/77
BASIC Interpreter Manual		30000-90026	14.75	6/76	8/78
FORTRAN Reference Manual		30000-90040	13.00	6/76	5/79
SPL Pocket Guide	#	32100-90001	2.45	11/76	
FORTRAN Pocket Guide	#	32102-90002	5.00	5/79	
BASIC Compiler Reference Manual	#	32103-90001	3.00	11/74	6/76
RPG/3000 Compiler Reference Manual	#	32104-90001	22.00	2/77	5/80
*RPG Utilities Reference Manual	#	32104-90006	@	2/81	
RPG Listing Analyzer	#	32104-90003	.50	2/77	
APL Reference Manual		32105-90002	35.00	1/79	
APL Pocket Guide		32105-90003	4.50	11/76	
COBOL Reference Manual	#	32213-90001	12.00	7/75	1/79
Using COBOL: A Guide for the COBOL Programmer	#	32213-90003	13.00	3/78	
*COBOL/II Reference Mnl.		32233-90001	19.00	12/79	7/80
COBOL/3000 to COBOL II /3000 Conversion Guide		32233-90005	4.00	12/79	

These manuals have been temporarily removed from circulation.

@ Price has not been established.

SOFTWARE PRODUCTS

Manual Title	V	Part Number	Price	Print Date	Up-dated
*DSG/3000 Manual		32250-90001	24.00	11/80	
*DSG/3000 Guide		32250-90002	4.25	11/80	
*TDP/3000 Reference Manual		36578-90001	30.00	4/81	
*Using TDP/3000		36578-90002	30.00	11/80	
*TDP/3000 Quick Reference		36578-90003	12.00	11/80	

DATA COMMUNICATIONS MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
Guidebook to Data Communications	#	5955-1715	3.00	1/77	
RJE/3000 Remote Job Entry (2780/3780 Emulator) Ref. Manual		30000-90047	12.75	11/79	
Communications Handbook		30000-90105	21.75	4/81	
HP 30010A Intelligent Network Processor (INP) Installation & Service Manual		30010-90001	10.50	10/79	6/80
HP 30010A/30020A Intelligent Network Processor Diagnostic Procedures Manual		30010-90002	4.25	10/79	6/80
HP 30020A Intelligent Network Processor (INP) Installation & Service Manual		30020-90001	6.25	10/79	6/80
HP 30032B Asynchronous Terminal Controller Instl. & Serv. Manual		30032-90004	14.00	1/74	7/76
HP 30055A Synchronous Single-Line Controller (SSLC) Instl. & Serv. Manual	#	30055-90001	8.50	12/77	4/79
*HP 30018A/30019A Add-On ADCC Installation Manual		30070-90023	@	3/81	6/81
Hardwired Serial Interface (HSI) Instl. & Service Manual		30360-90001	6.00	3/77	5/79

DATA COMMUNICATIONS MANUALS (continued)

Manual Title	V Part Number	Price	Print Date	Up-dated
*DS/3000 Reference Manual	32190-90001	19.00	3/77	5/81
DS/3000 to DS/1000 Reference Manual for HP 3000 Users	32190-90005	7.25	1/78	
*DSN/MRJE Multileaving Remote Job Entry Reference Manual	32192-90001	10.50	8/81	
*DSN/MTS Multipoint Terminal Software Reference Manual	32193-90002	8.00	8/81	
*DSN/IMF Interactive Mainframe Facility Reference Manual	32229-90001	20.50	9/81	

MANUFACTURING APPLICATIONS MANUALS

Manual Title	V Part Number	Price	Print Date	Up-dated
EDC/3000 User Reference Manual	32380-90001	20.00	3/78	4/79
EDC/3000 System Admin. Reference Manual	32380-90002	8.50	3/78	4/79
EDC/3000 Programmer's Reference Manual	32380-90003	20.00	3/78	
IOS/3000 User Reference Manual	32384-90001	25.00	3/78	
IOS/3000 System Admin. Reference Manual	32384-90002	11.00	3/78	
IOS/3000 Programmer's Reference Manual	32384-90003	23.50	3/78	
MRP/3000 User-Admin. Reference Manual	32388-90001	19.50	8/78	11/79

MANUFACTURING APPLICATIONS MANUALS (continued)

Manual Title	V Part Number	Price	Print Date	Up-dated
MRP/3000 Programmer's Reference Manual	32388-90002	13.00	9/78	
SPC/3000 User Reference Manual	32392-90001	11.00	4/79	
Master Production Scheduling and Rough Cut Resource Planning	32260-90001	17.00	7/80	
Maintaining Parts and Bills of Material	32260-90002	17.00	7/80	
Maintaining Routings and Workcenters	32260-90003	11.00	7/80	
Material Issues and Receipts	32260-90004	14.75	7/80	
Maintaining Work Orders	32260-90005	15.00	7/80	
Managing Inventory Balances	32260-90006	12.00	7/80	
Maintaining Purchase Orders	32260-90007	14.00	7/80	
Material Requirements Planning	32260-90008	7.25	7/80	
*Standard Product Costing	32260-90009	8.00	7/80	2/81
*System Customization	32260-90010	25.00	7/80	2/81
System Operation	32260-90011	8.00	7/80	
Materials Mgt/3000 Manual Set	32263A	110.00	7/80	

TRANSACTION PROCESSING MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
QUERY Reference Manual	#	30000-90042	9.00	6/76	5/79
*KSAM Reference Manual		30000-90079	14.50	5/79	5/81
*HP V/3000 Ref. Manual		32209-90001	14.50	1/80	2/81
HP V/3000 Entry Program		32209-90003	2.50	1/80	
Using HP V/3000		32209-90004	17.00	1/80	
IMAGE Data Base Management Reference Manual		32215-90003	13.75	9/79	3/80

EDUCATIONAL APPLICATION MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
Student Information System Reference Manual	#	32900-90001	13.00	9/74	8/76
Student Information System Technical Mnl	#	32900-90005	32.00	3/75	
Student Assignment System Reference Manual	#	32901-90001	15.50	8/78	
Student Assignment System Technical Manual	#	32901-90005	9.75	8/78	
College Information System Reference Manual	#	32902-90003	13.00	1/78	
College Information System Technical Mnl.	#	32902-90005	10.50	2/78	

ADDITIONAL MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
HP 3000 Series System Support Log		03000-90117	20.00	2/80	
HP 3000 CX to HP 3000 Series II Program Conversion Guide		30000-90046	3.50	6/76	
Guide to a Successful Installation	#	30000-90135	7.00	12/79	
Technical Writer's Survival Kit		30000-90171	2.50	7/79	
HP 3000 Computer System Site Planning Set (Encompasses the 2 manuals below)		30000-60029	15.75	6/80	
HP 3000 Computer System Site Planning and Preparation Guide		30000-90206	5.75	6/80	9/80
HP 3000 Computer System Site Planning Wkb		30000-90207	9.75	6/80	
Series 33 Installation Manual		30070-90021	5.25	10/78	1/80
Series 30/33 Diagnostic Manual Set		30070-60068	86.25	7/80	

ADDITIONAL MANUALS (continued)

Manual Title	V	Part Number	Price	Print Date	Up-dated
Series 30 Installation Manual		30080-90001	9.25	8/79	1/80
HP 2894A Card Reader Punch Operating Manual		30119-90009	11.50	10/76	
IBM System/3 to HP 3000 Conversion Guide	#	32104-90004	10.75	7/78	

MPE C MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
BASIC Interpreter Reference Manual		03000-90008	9.75	7/75	
Compiler Library Reference Manual		03000-90009	11.50	2/76	
Scientific Library Reference Manual		03000-90010	5.75	7/75	
Software Pocket Guide		03000-90126	4.60	7/78	
IMAGE Data Base Management Reference Manual		30000-90041	7.00	12/76	5/78

MPE C MANUALS (continued)

Manual Title	V Part Number	Price	Print Date	Up-dated
MPE Intrinsics Reference Manual	30000-90087	20.00	4/77	4/78
MPE Commands Ref. Mnl.	30000-90088	20.00	4/77	4/78
System Manager/System Supervisor Manual	30000-90089	12.50	4/77	4/78
Console Operator's Guide	30000-90090	11.00	4/77	4/78
General Information Manual (Series I)	30000-90091	9.25	4/77	
INDEX/3000 Reference Mnl	30000-90095	10.50	6/77	4/78
RJE/3000 (2780/3780 Emulator) Ref. Mnl. for Pre-Series II Systems	30130-90001	9.00	12/74	1/80
MPE System Utilities Reference Manual	32000-90008	2.05	10/75	
FORTRAN Reference Manual	32102-90001	10.00	3/76	
IBM 1130/1800 to HP 3000 FORTRAN Conversion Gd.	36995-90013	4.70	2/75	5/75