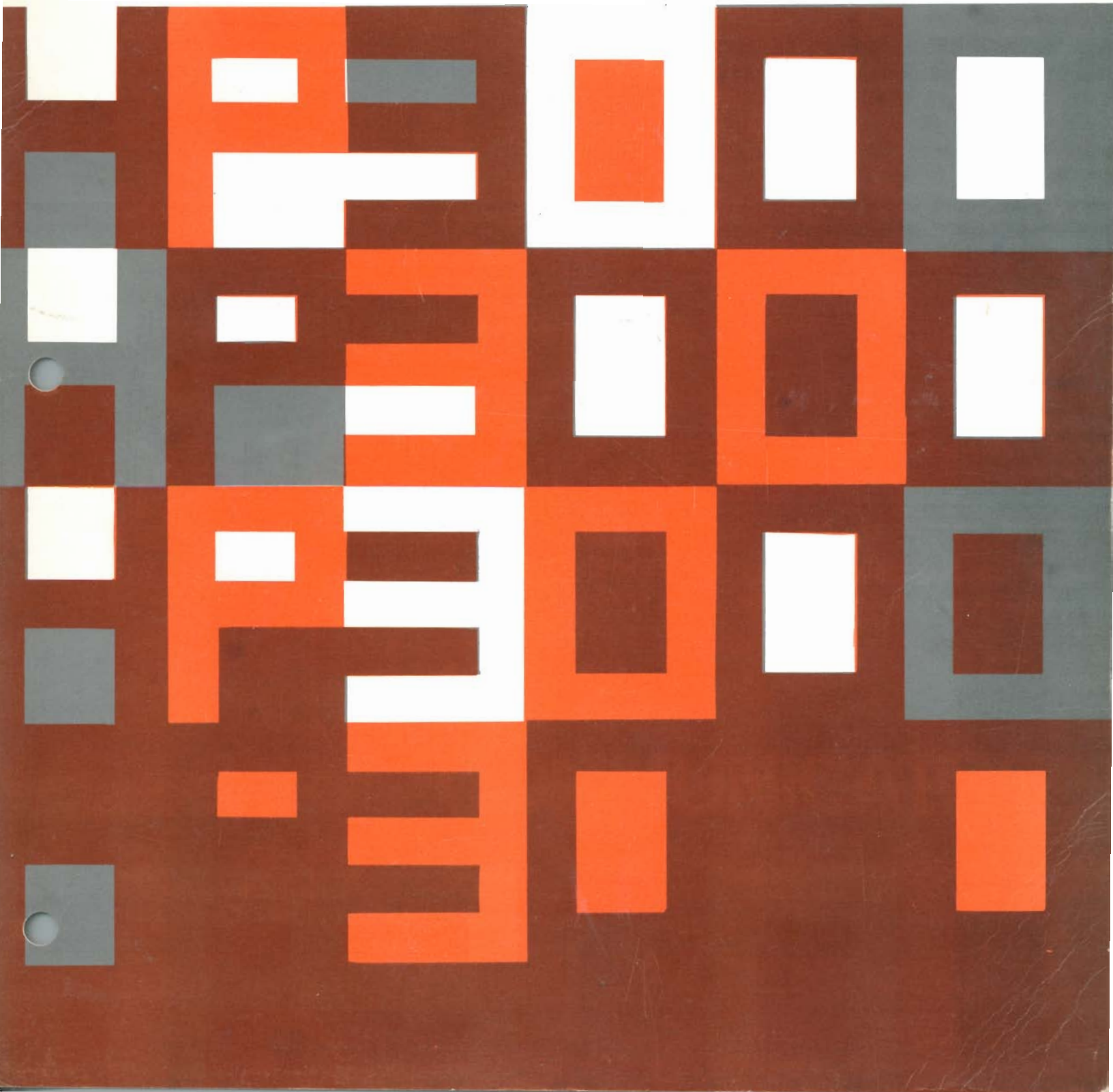
MPE IV 2052

HEWLETT PACKARD

COMMUNICATOR

ISSUE NUMBER 26



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Table of Contents

Editor's Note

This issue of the Communicator is intended to give you an idea of the strength of the contents of Installation Tape 2052. This product release is characterized by the power of the new HP 3000 Series 44 computer, as well as by the ease of use of several new facilities and the self-paced courses.

On pages 5 through 7, an outline of the changes to MPE summarizes the new features and enhancements that characterize MPE IV. The points mentioned in this list are discussed in more detail in other articles in the Communicator.

An overview of the HP 3000 Series 44, beginning on page 8, discusses software compatibility and the high reliabiliy of the Series 44, and concludes with information concerning the growth path for HP 3000 users to follow. A report on the performance of the Series 44 and MPE IV begins on page 10. Characteristics of MPE IV are described in an article beginning on page 14.

Interprocess Communication (IPC) is a powerful new facility that permits processes to communicate with one another. Beginning on page 18, the power and ease of use of IPC are outlined.

A new file system intrinsic, FREADBACKWARD, is described on page 21; a new utility, DPAN4, is also mentioned on page 21. An enhancement to the STREAM command is announced on page 22.

A comparison of the stack requirements of MPE IV and MPE III begins on page 23 and is followed by an explanation of the rigorous stack underflow checking algorithms implemented on the Series 44.

A new, easy-to-use graphics package, Decision Support Graphics/3000 (DSG), is introduced beginning on page 29. A new group of training products for HP 3000 users, the Self-Paced Learning Series, is detailed on page 31.

The new RPG utility, XSORT, is announced on page 32. The HPIB 2613, 2617, and 2619 line printers now provide Series 30/33/44 systems with high-speed printing capabilities; this development is explained on page 34.

There are several new enhancements to IML/3000; these software changes are reported beginning on page 35. The syntax of the "TRACE" parameter of the RJE/3000 #RJLINE command has been revised; the new syntax is discussed in an article beginning on page 38.

Beginning on page 42, some tips for debugging programs compiled with the new COBOL II/3000 compiler are provided. A discussion of new commands and other changes to the Segmenter begin on page 54.

The Documentation Section, beginning on page 63, reports recent documentation activity. The latest Catalog of Customer Publications begins on page 75.

Editor
COMMUNICATOR 3000

Summary of Changes for MPE IV

The following is a list of all new features, enhancements, and changes to MPE that are included in MPE IV. The list includes features which are visible as well as features which are transparent but still affect the user. These items are discussed in more detail in the articles in this issue of the Communicator.

1. The MPE IV Kernel

   This includes a totally re-written memory manager, cpu scheduler, and disk scheduler. Users will note the following changes and features:

   * Changes to the SYSDUMP/INITIATOR dialog to accommodate deleted and added tables and features.

   * Multi-spindle virtual memory. This allows for both a more even system load on the disks and a greatly expanded amount of virtual memory.

   * Support of up to 2**32 words of memory. Obviously, only have an immediate need for 4 megabytes.

   * Inclusion of look-ahead seek on Series II/III systems.

   * Measurement interface subsystem.

   * Deletion of the :QUANTUM command and addition of :TUNE command as its replacement.

2. New File System I/O

   This includes a greatly speeded up file system by elimination of unnecessary EXCHANGEDBs. Also added are as follows:

   * An FREADBACKWARD intrinsic for 7970E/7976A devices.

   * Ability to append to variable record format files.

   * A new file type, circular files, which can be used as history files, for example. (The last n records written are saved.)

* An INTERPROCESS COMMUNICATION facility. This allows for efficient process to process communication under the file system including across DS lines.

3. Speeded up KSAM

This includes deletion of unnecessary EXCHANGEDBs and more efficient handling of buffers during multiaccess to KSAM files.

4. Enhanced DEBUG Features

This includes the following additional features to DEBUG.

* Dumping of memory in instruction format.

* Conditional breakpoints.

* Cross-pin breakpoints (available to privileged users only).

5. Enhanced RESUMESPOOL Command

This includes the ability to backspace, forwardspace, and rewind spoolfiles on a page or file basis.

6. Speeded up SPOOK

This includes the ability to list any part of a spoolfile more quickly because the file system now allows random access to a spoolfile. Thus, SPOOK does a binary search, rather than a serial search for specific line numbers.

7. Redirection of $STDIN/$STDLIST at :RUN time

The user may now specify alternate paths for $STDIN and $STDLIST for run time programs using the :RUN command.

8. New Dump Analyzer for MPE IV -- DPAN4

DPAN4 has been written to analyze the greatly different tables structures inside of MPE IV. Additionally, selective analysis of individual tables can be made by setting ;PARM=10 on the :RUN command.

9. Additional Stack Requirements

Various intrinsics require more stack in order to execute. Typically, the file system intrinsics take 120-160 more words, and data segment intrinsics take 5-30 more words.

10. Addition of a System Logging Type 16

This logging type logs underflow simulation of programs on Series 44.

11. Ability to :STREAM Files with Execute Access

This allows a creator of a file to put secure information into a stream file and disallow allow read access to the file, yet others may still stream the file.

12. XSORT Facility for RPG/3000

This makes a supported utility out of the previously unsupported RPG utility, SORT3.

13. Support of 2613, 2617, 2619 on HPIB systems

14. Enhanced IML/3000

This includes the addition of the ATTRLIST intrinsic for returning the attribute characters within any portion of the internal screen image. The DISPLAY command has been enhanced in the IML manager to display the usage and availability of all IML devices. IML is also available on HPIB systems across the INP.

15. Changes to TRACE in RJE

The trace portion of the RJLINE command has been revised.

16. Segmenter enhancements

The following commands have been added to the Segmenter: CLEANUSL, CLEANSL, COPYSL, and COPYUSL.

Introducing ...The HP 3000 Series 44

by John Chisholm, Computer Systems Division

It supports up to four megabytes of main memory...nearly two GIGAbytes of disc storage...and provides transaction throughput up to twice that of the HP 3000 Series III with MPE III...It's the HP 3000 Series 44!

The HP 3000 Series 44 is the new high-performance member of the HP 3000 family. It has a new CPU whose speed is 60% faster than that of the Series III; uses new, faster memory; and supports four megabytes--twice the amount of memory supported in the Series III. Including both point-to-point and multipoint, it supports up to 96 terminals. And it uses MPE IV, which provides an improved memory manager and a more efficient MPE file system. But most of all, the HP 3000 Series 44 provides the best price/performance of any HP 3000 system offered yet. The base system includes 1 Mb memory and two General I/O Channels (GICs) bundled into the system, with one 50 Mb disc, 1600 bpi tape drive, ADCC-Main and 2621A console ordered separately.

Software Compatibility

The Series 44 comes with the same Fundamental Operating Software that is bundled with every HP 3000 system. It runs any of the five HP 3000 languages--COBOL(II), FORTRAN, BASIC, RPG, and SPL--and is fully compatible with HP-Distributed Systems Network (HP-DSN). DS/3000, RJE/3000, MRJE/3000, IML/3000, and MTS/3000 (Multipoint) are all supported on the Intelligent Network Processor (INP) on the Series 44--meaning high performance data communications, especially for Multipoint. In addition, TDP/3000, DSG/3000, Materials Management/3000, and Flexible Disccopy/3000 (used with the standalone Model 9895A flexible disc drive) are all available on the Series 44.

The Series 44 is software compatible with the Series 30, 33, II, and III. Programs developed on other HP 3000s can be transferred to the Series 44 with recompilation rarely being required. This means your present investment in HP 3000 applications software will be preserved by the Series 44.

Highest Reliability

Series 44 System Processor Unit reliability is so high that SMMC's are actually lower than those of the Series 30! This high reliability is made possible by a diagnostic subsystem built into the Series 44 mainframe called the Control and Maintenance Processor (CMP). The CMP performs all of the same functions of

the 2649E System/Maintenance Console on the Series 30 and 33, and then some! Its self-tests and diagnostics are not only more thorough than those of the 2649E, they're also easier to use! The CMP means that any HP terminal--including the 2621A--can act as Series 44 system console...a real savings!

The Series 44's high reliability has enabled HP to offer an industry first: Guaranteed Uptime Service. This new service guarantees up to 99% uptime for the Series 44. Guaranteed Uptime Service, available only with the Series 44, is ordered through the GMMC (Guaranteed Monthly Maintenance Charge), a new alternative to SMMC (Standard Monthly Maintenance Charge) or BMMC (Basic Monthly Maintenance Charge). GMMC is priced at less than 7% of the Series 44 System Processing Unit price--a small premium for guaranteed 99% uptime!

Growth Path For All HP 3000's

The Series 44 provides a compatible growth path for all existing HP 3000 users. For the Series 30 and 33, a simple field exchange of System Processor Units is required to upgrade to the Series 44. All peripherals are directly transferable. Regardless of how long ago you bought a Series 30 or 33, we'll give you up to a 50% return credit on your Series 30 or 33 System Processing Unit towards the purchase of the Series 44. If you have one of the other HP 3000's and you wish the added performance of the Series 44, you can also return your System Processing Units to HP for credit towards the purchase of the Series 44.

HP 3000 Series 44 and MPE IV Performance

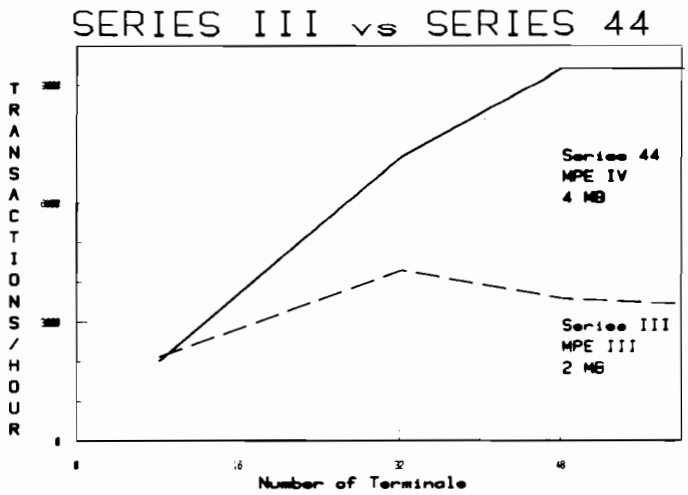
by John Chisholm, Computer Systems Division

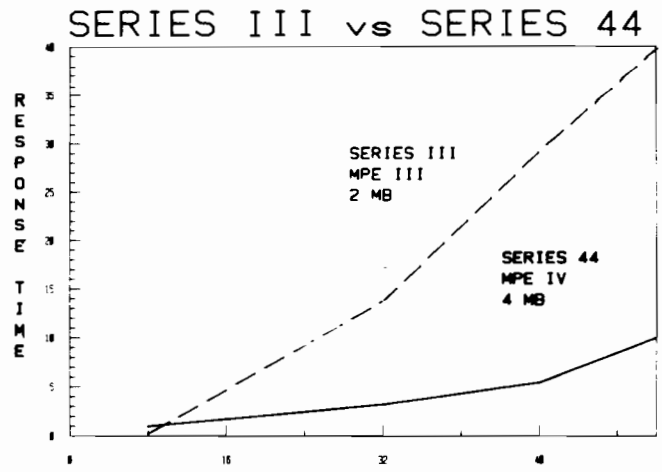
The information contained in the following graphs presents the performance of different HP 3000 Computer Systems in a general-purpose EDP environment. The data shown is intended to serve as guidelines for typical performance of the HP 3000 family of compatible business systems. Performance is measured in terms of transaction throughput per hour and average response time for on-line terminals. Comparisons are shown between MPE III and MPE IV versions of the MultiProgramming Executive operating system. The test environment used is detailed at the end of this article.

Actual HP 3000 system performance will depend upon both the configuration and the application. Configuration factors affecting performance include the System Processor Unit, the size of main memory, the system disc(s) configuration, and terminal and data communications usage. Actual performance in any particular configuration will depend upon the application with its mix of such factors as data entry, inquiry, program compilations, file sorts, batch processing, and data communications.

Highest Performance HP 3000 Series 44

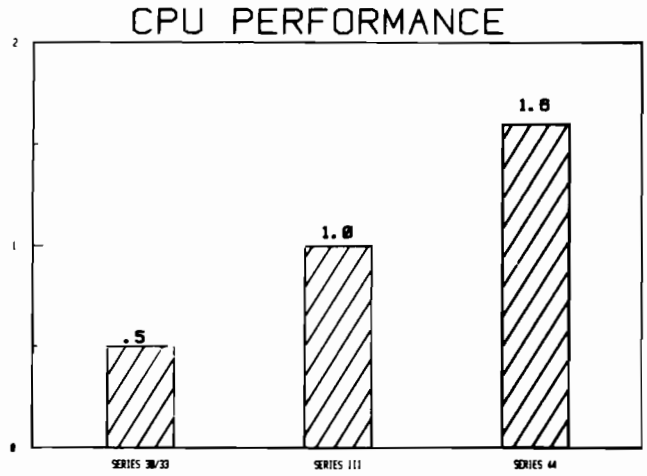
The HP 3000 Series 44 is the highest performance member of the HP 3000 family. The graphs below compare throughput and response time for the Series 44 with MPE IV and its maximum memory (4 Mb) with the HP 3000 Series III with MPE III and its maximum memory (2 Mb).





Conclusion: For this application, the Series 44 continued to provide transaction throughput and response time gains at and beyond the terminal load where the Series III had achieved its maximum throughput.

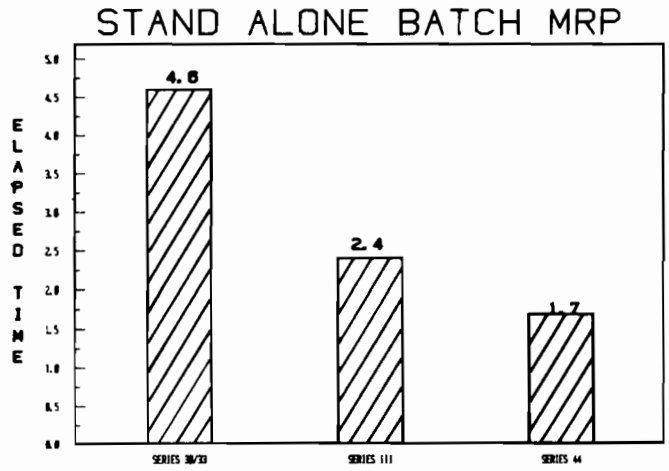
The graph below shows the throughput of the HP 3000 systems with identical memory sizes running stand-alone programs which exercise the CPU. The results are based on total program completions per hour over a dozen different programs.



Conclusion: The Series 44 CPU performance is 1.6 times the Series III and 3.2 times the Series 30 and 33.

Standalone Batch Processing

A single Material Requirements Planning batch job was run on the HP 3000 systems using Materials Management/3000 software. The graphs below indicate the relative performance of the three systems, based on the elapsed time required to process the job on each system.



Conclusion: The Series 44 handled the MRP job in 30% less time than the Series III, and about 60% less time than the Series 30 and 33.

Test Environment

The application environment in which the transaction processing tests were conducted consisted of the following mix of activities:

1. Program Development--1200 line COBOL programs were entered through the EDITOR followed by program compilation and preparation;

2. V/3000 and IMAGE/3000--a COBOL program verified entered data against an IMAGE data base, then updated that data base; the data base consisted of 50,000 records organized into ten detail and 18 master data sets; V/3000 data entry activity consisted of a series of three screens;

3. Data Inquiry--QUERY/3000 was used for online data base inquiry and reporting; the data base used was the same as described above; the inquiries were FIND's and REPORT's;

4. MPE Commands--commands such as LISTF, HELP, and SHOWJOB were executed;

5. Batch--A COBOL program read 10,000 records from a file and sorted them on two keys; a COBOL report program produced a sorted report from the data base, printing summary totals for each sort field; the file was sorted twice and two reports generated.

As the number of terminals varied, the mix of terminals was held fixed. The following mix of on-line terminals was used in the test environment: COBOL program development, 10% of terminals; V/3000 and IMAGE/3000 updating, 75% of terminals; QUERY/3000, 12% of terminals; MPE Commands, 3% of terminals.

Note: All tests were conducted using 2400 bps terminals. Data entry with V/3000 was done in block mode. Performance results shown here apply to the test application mix only, which is intended to be indicative of a general purpose EDP environment. Transaction throughput and response times on HP 3000 systems will depend upon your specific configuration and application.

Introducing MPE IV

by Bob Stamps, Computer Systems Division

MPE IV is a step forward in system resource utilization and performance. Computer Systems Division has improved process scheduling, memory management, virtual memory allocation, disc I/O request scheduling, and the file system. Other enhancements involve the assignment of $STDIN and $STDLIST, the RESUMESPOOL console operator command, and the DEBUG facility.

PROCESS SCHEDULING

The process scheduling algorithms of MPE III allowed low urgency batch jobs to increase in priority so that they competed with high urgency interactive transactions during periods of heavy terminal activity. This had a tendency to decrease terminal response at critical times. The new scheduler for MPE IV tends to filter out long transactions in each scheduling queue so that they will not compete with short transactions (unless competition is desired).

MEMORY MANAGEMENT

The MPE memory manager has been improved for MPE IV in a number of ways. For example, once the MPE III memory manager began to satisfy a process's request for memory, it had to wait for the requested segments to arrive in memory before starting on another request. The MPE IV memory manager, on the other hand, has to wait only long enough to set up the requests for the I/O system before starting on another request. This "parallel" request handling feature increases the efficiency of memory allocation.

The memory manager of MPE III used a first fit algorithm to find free space in memory for a requested segment. The MPE IV memory manager uses a best fit approach. This makes it more likely that large free areas will be available for large segments. MPE III used a program working set algorithm to create room in memory once it was determined that enough free space did not exist to satisfy a request. Only one working set table entry was used per executing program to keep track of which segments were needed in memory by that program. Since more than one process could be running the same program, it was possible for segments to be marked as candidates for possible removal from memory even though one of the processes running the program needed the segments to run efficiently. MPE IV uses a replacement algorithm

which selects candidates for memory overlay by looking at how often segments in memory are used by any process on the system. This reduces the probability of removing a heavily used segment from memory.

In MPE III, once a data segment had been selected as an overlay candidate, an anticipatory write request was placed in the I/O request queue so that the copy of the data on disc would be updated in case the segment was chosen to be overlaid by a more important segment. The request was placed in the queue in front of all other requests for the disc, pre-empting file system disc I/O. Often a write would complete even though the overlay candidate was never actually chosen for overlay. In MPE IV this wasted I/O activity is minimized by placing the anticipatory writes of overlay candidates at very low priorities.

The MPE III memory manager performed memory compaction when segments were returned to the free list to combine small free areas into larger free areas, thereby increasing the probability that large memory requests could be satisfied. In MPE IV, memory compaction is also done during what would otherwise be idle time. Since idle time sometimes occurs because of an inability to find space for a process's segments, this compaction tends to increase the efficiency of the memory manager.

VIRTUAL MEMORY ALLOCATION

MPE III allowed virtual memory to be configured only on logical device 1. Because the directory and system SL also resided on this volume, contention for ldev 1 was very high and was a bottleneck on many systems. MPE IV allows virtual memory to be configured on any or all system domain volumes. This tends to even out the disc queue lengths among the system domain volumes and thereby reduce contention for ldev 1. In addition, the maximum virtual memory size is greatly increased.

DISC I/O REQUEST SCHEDULING

In MPE III, I/O requests for a disc device were handled on a first-come, first-serve basis without regard for the priority of the processes requesting the I/O (with the exception of memory manager disc I/O). This meant that I/O for a process scheduled in the ES queue could hold up disc resources for a process scheduled in the CS queue. In MPE IV, disc I/O for each disc device is handled on a process priority basis. This allows disc resources, not only CPU resources, to be allotted depending upon process urgency.

FILE SYSTEM

The MPE III File System used a method of control block handling which required the use of many EXCHANGEDB instructions. These instructions were very time consuming. The MPE IV File System places control blocks momentarily on the user's stack so that the EXCHANGEDB instructions are not necessary. (See article in this issue of the COMMUNICATOR.) The result is faster file access. In addition, much of the code for the File System has been optimized.

In addition, enhancements to some File System intrinsics have greatly simplified communication between separate processes. See the article on Interprocess Communication in this issue.

REDIRECTION OF $STDIN/$STDLIST

$STDIN and $STDLIST may now be redirected at run time through utilization of the following form of the RUN command:

```
:RUN PROGRAM; STDIN=FILENAME; STDLIST=FILENAME [,NEW]
```

If nothing is specified on the right hand side of the "=" sign, the system default files are used. If a new file name is specified on the right hand side of the "=" sign, the file would be defined as fixed length ASCII, 132 bytes with a maximum file size of 1023 records.

RESUMESPOOL COMMAND

The RESUMESPOOL command may be used to backspace and forwardspace files or pages, or to resume printing at the beginning of a file which was suspended during printing (or at the beginning of the last unpurged extent). The format is as follows:

```
:RESUMESPOOL ldev; BACK     (nnn FILES)
                   FORWARD  (nnn PAGES)
                   BEGINNING
```

ldev — logical device number of printer

BACK — backspace nnn files and begin printing, or backspace nnn pages and begin printing

FORWARD — forwardspace nnn files or pages and begin printing

BEGINNING resume printing at the beginning of the first unpurged extent or at the beginning of the file, whichever applies

DEBUG

The DEBUG facility has been enhanced for MPE IV to allow two new options. First of all, a word of code can now be displayed mnemonically using the C option of the D (DISPLAY) command.

```
EXAMPLE: ?D P,C
          -----

         P+0 045403  LOAD Q+3,X
```

In addition, conditional breakpoints may now be set so that a breakpoint will take effect only if a certain condition is true.

```
EXAMPLE: ?B 3.2 : DB+3=5
          --------------
```

The breakpoint will exist only if DB+3=5.

Interprocess Communication (IPC)

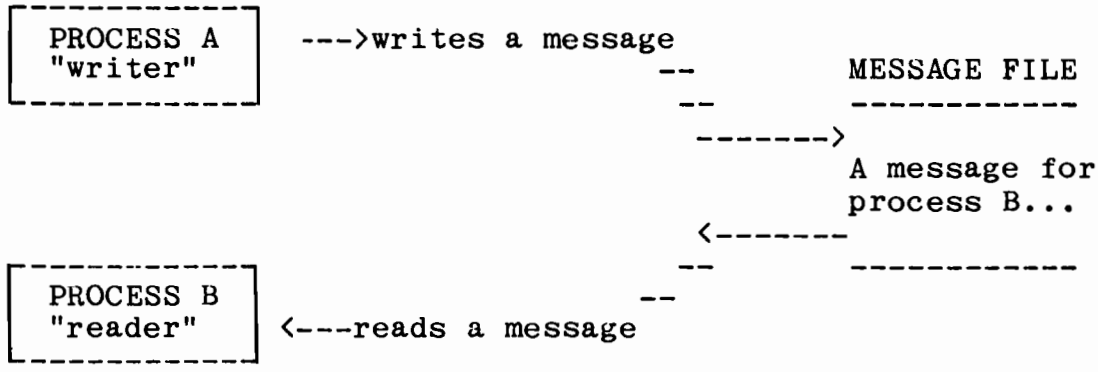
The new Message File facility provides an efficient method for processes to communicate with each other.

by Adrienne Bresso, Computer Systems Division

Since its introduction, MPE has offered a variety of tools to allow users to perform process to process communication and control. However, these mechanisms have proven to be too restrictive to allow applications to take full advantage of MPE's capability.

Interprocess Communication is a new communication facility that removes many of the existing restrictions. It allows communication between multiple user processes, not necessarily in the same process tree. In other words, any process may communicate with any other, subject to the normal File System security provisions.

To implement IPC, a new file type is used, referred to as a MESSAGE file. MESSAGE files are disc files accessed with the standard File System Intrinsics that are used to access standard sequential files. The advantage of this is that it is not necessary to become familiar with a new set of intrinsics. Also, it makes it possible for existing programs to redirect their input and output to MESSAGE files.



The MESSAGE file can be thought of as a (first-in/first-out) FIFO queue of messages where each FWRITE appends a message to the end of the file, and each FREAD reads and deletes the first message in the file. The MESSAGE file is unidirectional; that is, a reader of the MESSAGE file desiring to talk to a writer of the

MESSAGE file must do so via another MESSAGE file in which their reader/writer roles are reversed.

MORE GENERAL ADVANTAGES OF IPC:

TIMEOUTS:

This is an optional feature that permits detection of deadlocks. For example, if there is a reader process attempting to read from an empty file, he may be waited with a time limit. If no writer process has written to the MESSAGE file within the specified time limit, the waiting reader process will be taken off of the waiting list for a message. In this way, there is no danger of the reader process waiting for a message forever.

OBITUARY NOTICES:

When a writer process to a MESSAGE file terminates, a CLOSE record is automatically written to the file. The CLOSE record contains information about what process just stopped writing and closed the file. This information can optionally be given to the reader.

GLOBAL MULTIACCESS FILES:

When the global multiaccess option is requested, processes located in different jobs or sessions may open the same message file.

APPEND ACCESS TO VARIABLE-LENGTH FILES:

Variable-length files may be opened with append access. It is not necessary to have fixed-length records of the maximum possible size, so space is conserved.

CHOICE OF NOTIFICATION:

MESSAGE file completions can be handled with WAIT or NOWAIT I/O, neither of which require Priviledged Mode capability while using IPC.

APPLICATIONS MAY SPAN COMPUTERS:

Combining MESSAGE files with DS/3000 allows processes to be remote from their files as well as from each other.

FLEXIBILITY:

MESSAGE files' attributes can be changed with the :FILE command.

CIRCULAR FILES:

Circular files are new features related to IPC. They are wrap-around structures which behave as standard sequential files until they are full. As records are written to a circular file, they are appended to the tail of the file; when the file is filled, the next record added causes the block at the head of the file to be deleted and all other blocks to be logically shifted toward the head of the file. Circular files may not be simultaneously accessed by both readers and writers. When the file has been closed by all writers, it may be read; a reader takes records from the circular file one at a time, starting from the head of the file.

Circular files are particularly useful as history files, when a user is interested in the information recently written to the file and is less concerned about earlier material that has been deleted. These history files are frequently used as debugging tools: diagnostic information may be written to the file, and the most recent and relevant material can be saved and studied.

As you can see, IPC is a potentially powerful facility. It is not meant to take the place of existing process to process communication tools but rather to enhance them.

Freadbackward - - A New File System Intrinsic

by Adrienne Bresso, Computer Systems Division

FREADBACKWARD is a new intrinsic that permits access to the Read Reverse capability of the HP7970 Magnetic Tape drive.

FREADBACKWARD can be used to recover tape errors and can be of great use to the user who is handling his/her own I/O management and data recovery. FREADBACKWARD has two substantial restrictions:

1. It applies to Magnetic Tape on the Series 30/33 only.

2. The tape must be accessed NOBUF.

The intrinsic syntax is as follows:

```
lgth:=FREADBACKWARD(filenum,target,tcount);
```

where lgth, filenum, target and tcount have the same definition as for FREAD. Refer to the MPE INTRINSICS manual for further information.

DPAN4 - A program that produces an analytical formatted listing of main memory for MPE IV.

by Adrienne Bresso

DPAN has been modified to correctly format main memory dumps taken on systems running on MPE IV. THe new dump analyzer program is called DPAN4. Refer to the MPE System Utilities manual for further information.

Enhanced STREAM Command

Many HP/3000 customers have expressed concern about the limited security provided by the STREAM command. In the past, users had to have READ access to a file in order to STREAM it. This implied that the creator of a job file could not put secure information, such as passwords and lockwords, within the file and still make it available for general use. However, an MPE IV enhancement allows users to STREAM any file for which they have READ or EXECUTE access. Thus, the creator of a job file can keep the contents of the file secure by allowing only EXECUTE access to the users of the file.

Stack Requirements of MPE IV vs MPE III

By Bob Stamps, Computer Systems Division

Computer Systems Division has made performance-improving changes to the File System in MPE IV. For instance, the File System now copies the Access Control Block (ACB) to the top of the stack after calls to File System intrinsics. For this reason, programs written on MPE III have greater stack space requirements (above DB) when run on an MPE IV system.

The table below lists the most frequently used File System and Data Manipulation intrinsics and the approximate increase in stack requirements from MPE III to MPE IV, according to our tests.

INTRINSIC	APPROX. INCREASE IN STACK SIZE (WORDS)
ALTDSEG	13
DLSIZE	9
DMOVIN	3
DMOVOUT	0
FCLOSE	11
FCONTROL	94
FFILEINFO	162
FOPEN	167
FREAD	126
FREADDIR	1
FREEDSEG	25
FWRITE	118
FWRITEDIR	162
GETDSEG	30
PRINTFILEINFO	135
ZSIZE	30

Because of this stack size increase you may need to increase the initial stack size and/or the maxdata value for some of your programs. You may do this by re-PREPing your program with new stack size and/or maxdata values or, if source is not available, by specifying them in the :RUN command each time the program is executed.(See COMMANDS MANUAL for description of maxdata and stacksize parameters.)

Since the maximum stack size allowed in any user stack on your system is configurable (see System Manager/System Supervisor Manual), you may wish to change the value on your system because of the MPE IV changes. The standard stack size (the initial stack size assigned when the user does not specify a size at prep time) is also configurable and you may wish to reconfigure it also.

If some of your programs were approaching the absolute maximum stack size allowed on the system (31232 words) while running on MPE III, it may be necessary to request that these programs run without using the PCBX area of the stack for holding control blocks by specifying NOCB each time the programs are run on MPE IV. (See COMMANDS MANUAL.)

Stack Underflow Simulation

by Adrienne Bresso, Computer Systems Division

Users upgrading to an HP3000 Series 44 may experience a compatibility problem because of the more stringent stack underflow checking algorithms implemented on the Series 44.

The HP3000 Series 44 system has changed the definition of stack underflow. Stack underflow is now defined to be S < Q. This will improve system reliability by protecting stack markers from unwarranted access.

However, a problem exists in that the previously existing COBOL and BASIC compilers emit code for certain language constructs which violate the new definition. Object code produced by these compilers will no longer execute in a few cases. Five constructs which cause stack underflow are:

COBOL

Construct: DIVIDE with REMAINDER Code: DDUP when S=Q+1

Violation: Operand is at Q and Q+1.

Construct: PCAL to library procedures Code: DDUP when S=Q

Violation: Operand is at Q-1 and Q.

Construct: PCAL to library procedures Code: DDUP when S=Q

Violation: Operand is at Q.

BASIC

Construct: CHAIN Code:
```
LRA Q+0   when S=Q+1
XCH,SUB
DUP,NOP
SETR Q
```

Violation: XCH fails because operand is at Q and Q+1. Subsequent instructions would also fail if allowed to continue without simulation.

Construct: CALL

Code: LRA Q+0 when S=Q
SETR S
SETR Q

Violation: SETR Q fails because operand deletion causes S to fall below old value of Q at execution of SETR Q.

The COBOL and BASIC compilers have been modified so that the offending code is no longer produced. COBOL is updated in 2028. BASIC is updated in 2052. This will allow newly compiled programs to execute successfully but does nothing for older object code. The solution to this problem is to simulate the offending instructions in software via the FIRMWARESIM mechanism. FIRMWARESIM is used to simulate the EXTENDED INSTRUCTION SET whenever the hardware for those instructions is not present. The microcode checks for the presence of this hardware whenever one of the EIS op codes is encountered. If present, control is passed to the EIS microroutine and execution continues normally. If absent, an unimplemented instruction trap is generated and ININ is entered. The UNIMPLEMENTEDINSTRUCTION trap procedure validates the instruction and invokes the proper procedure in FIRMWARESIM.

The procedures in FIRMWARESIM utilize the stack markers for the trap and procedure call to fetch the operands and set the proper state for the return to the user procedure. Once the simulation is completed, control is returned directly to the user procedure, i.e., the trap procedure is bypassed.

It is desirable for the user to be notified of the simulation because a small amount of performance may be lost. This is accomplished by the creation of a new LOG file record.

The stack underflow simulation procedure flags the event in the PCBX on each occurrence. At process termination, a log record is posted in the system log file. The record contains the fully qualified name of the violating program file. Once a program has been identified as one experiencing stack underflow simulation, it can be corrected by simply recompiling the program (2028 version for COBOL and 2052 for BASIC).

SPL STACK UNDERFLOW SIMULATION

Although the intent of the stack underflow simulation is to allow COBOL and BASIC programs to execute as they have on older models of the HP3000, SPL programs could possibly encounter the same problem. This will only occur if the SPL program accesses words at or below Q with a stackop. The only way to do this is with an ASSEMBLE statement. Code of this type is generally easy to modify so that the trap can be avoided. For example, a DUP with S=Q

can be replaced with a LOAD from S-0.

Simulation will be performed for SPL programs if the code matches one of the COBOL or BASIC generated code sequences and the conditions for simulation are met, ie., the program is in user mode, simulation is enabled, and the processor type is Series 44 . The name of the program file will be logged in the system log file as for COBOL and BASIC programs.

LOGGING

A 16th event has been added to the system log facility. This new record type is defined as a general purpose program file event log. Special circumstances pertaining to program files, such as the stack underflow simulation, will be logged under this format. A sub-type field is provided allowing various kinds of events to be logged under this one format.

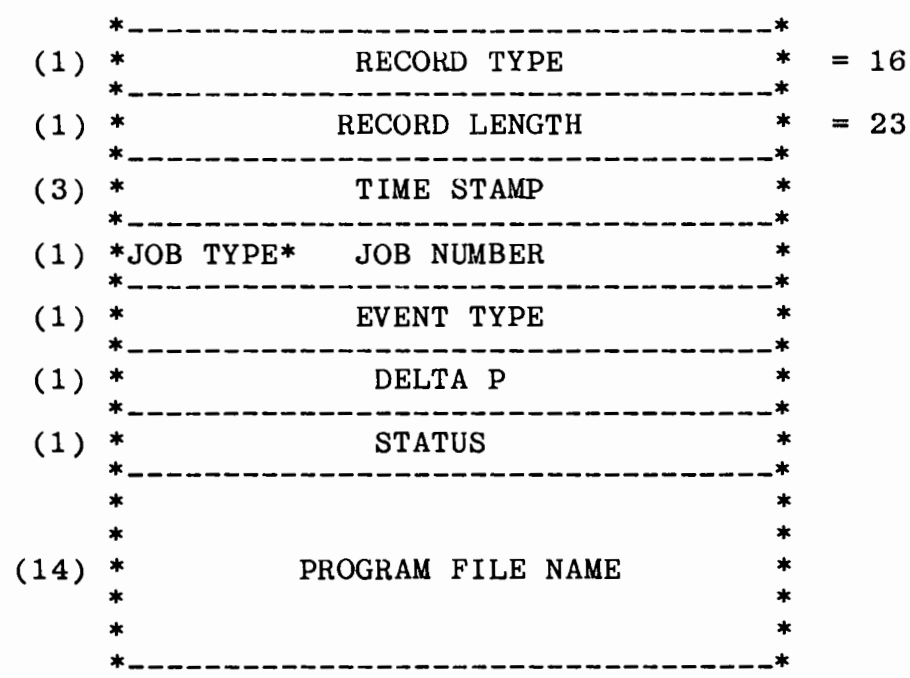
SYSDUMP LOG CONFIGURATOR

The new SYSDUMP logging status message appears as follows:

```
LIST LOGGING STATUS? y
TYPE        EVENT           STATUS
  1    LOGGING ENABLED        ON
  2    JOB INITIATION         OFF
  3    JOB TERMINATION .      OFF
  4    PROCESS TERMINATION    ON
  5    FILE CLOSE             OFF
  6    SYSTEM SHUTDOWN        OFF
  7    POWERFAIL              ON
  8    SPOOLING               OFF
  9    LINE DISCONNECTION     ON
  10   LINE CLOSE             OFF
  11   I/O ERROR              ON
  12   VOLUME MOUNT           OFF
  13   VOLUME SET MOUNT       OFF
  14   TAPE LABELS            OFF
  15   CONSOLE                ON
  16   PROGRAM FILE EVENT     ON
```

Enabling and disabling the PROGRAM FILE EVENT is done in the same manner as for the other events.

LOG TYPE 16 RECORD FORMAT



Notes:

The EVENT TYPE number for stack underflow simulation is 0. No other EVENT TYPES have been defined.

The DELTA P and STATUS fields are not used for stack underflow simulation.

LISTLOG2 OUTPUT

The output from LISTLOG2 for stack underflow simulation is as follows:

```
TYPE        EVENT TYPE              PROGRAM FILE    STATUS DELTA P

PRGM  STACK UNDERFLOW SIMULATION  NAME.GROUP.ACCT   N/A    N/A
```

Editor's Note: Decision Support Graphics/3000 is not on Installation Tape 2052. This facility is now available on a special product tape.

Introducing HP Decision Support Graphics/3000

by Jutta Kernke and Carla Klein, General Systems Division

Hewlett-Packard's Decision Support Graphics/3000 (HP DSG/3000), a business computer graphics software product for the HP3000 family, was introduced in September. HP DSG/3000 helps users design, save, and produce line graphs, horizontal and vertical bar charts, pie charts, and scattergrams. All graphs can be displayed on the 2647A and 2648A graphics terminals, or can be produced on the 7221A/B/S (RS-232) or 9872A/B/S (HP-IB) four-color plotters, the 7225A single-pen plotter, and the 7245B printer/plotter.

The product consists of an interactive program, GRAPH, and a comprehensive set of high-level procedures that can be called from COBOL, BASIC, FORTRAN, and SPL. GRAPH uses "fill in the blanks" menus to accept the users' selections and offers a built-in "Help" facility at the touch of a function key. Using GRAPH, charts can be designed on the 2641A, 2645A, 2626A, 2647A, or 2648A terminals.

Data for HP DSG/3000 charts can come from any columnar data file created by the EDITOR, V/3000, QUERY, a user-written application program, or can be entered through the terminal keyboard. Chart specifications and data are kept separate, so data can be updated and modified without changing the specifications. In addition data can be automatically scaled, and arithmetic and statistical functions can be applied to it before plotting. While many options and enhancements are available to the user, sensible defaults can always produce an aesthetically pleasing chart. Defaults include centering of titles, bar shading, line-dot-dash patterns, and overall balance.

DSG/3000 is designed with the periodic user in mind. Once a graph has been created, it can be saved in the chart file and reused on different or revised data. With an interactive option, DSG/3000 will produce a set of multi-color charts, entirely unattended, on the HP 7221 or 9872 S-model plotters.

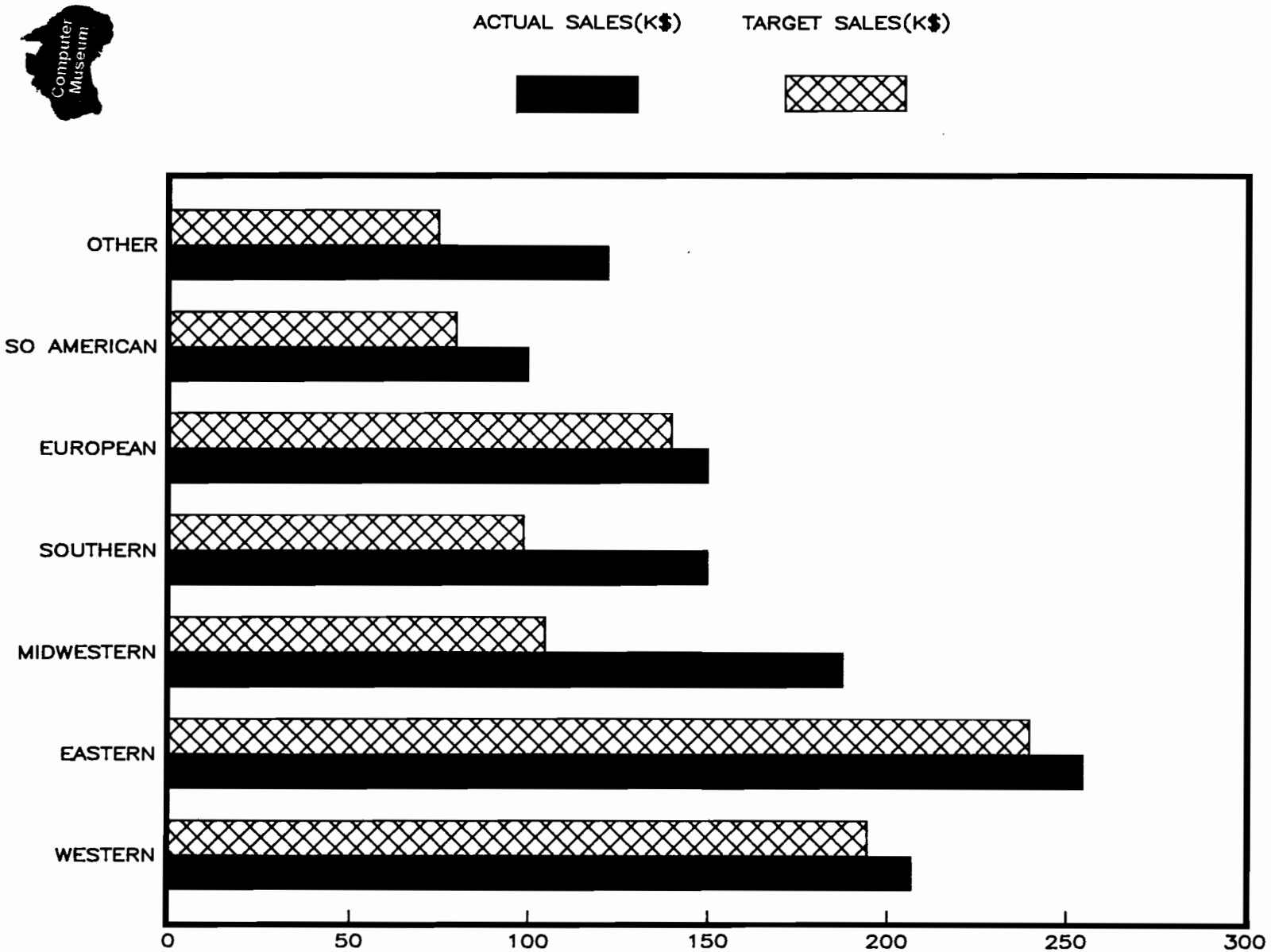
Two customer training courses are available for DSG/3000. A self-paced course, part number 22833A, is offered for the non-technical interactive user. For the experienced application programmer, there is a three-day classroom course which includes

the interactive training; part number 22834A.

DSG/3000 is ideal for producing accounting reports, statistics, sales forecasts, and production plans, and for tracking expenses. Rather than working with thousands of pages of numerical data which need to be sorted and studied, HP DSG/3000 offers an effective way to quickly interpret that information. Contact your local sales office for a demonstration of all of the capabilities of HP Decision Support Graphics/3000.

SALES BY REGION FOR 1979

ACTUALS VS. TARGETS



P/N: 5953-4076

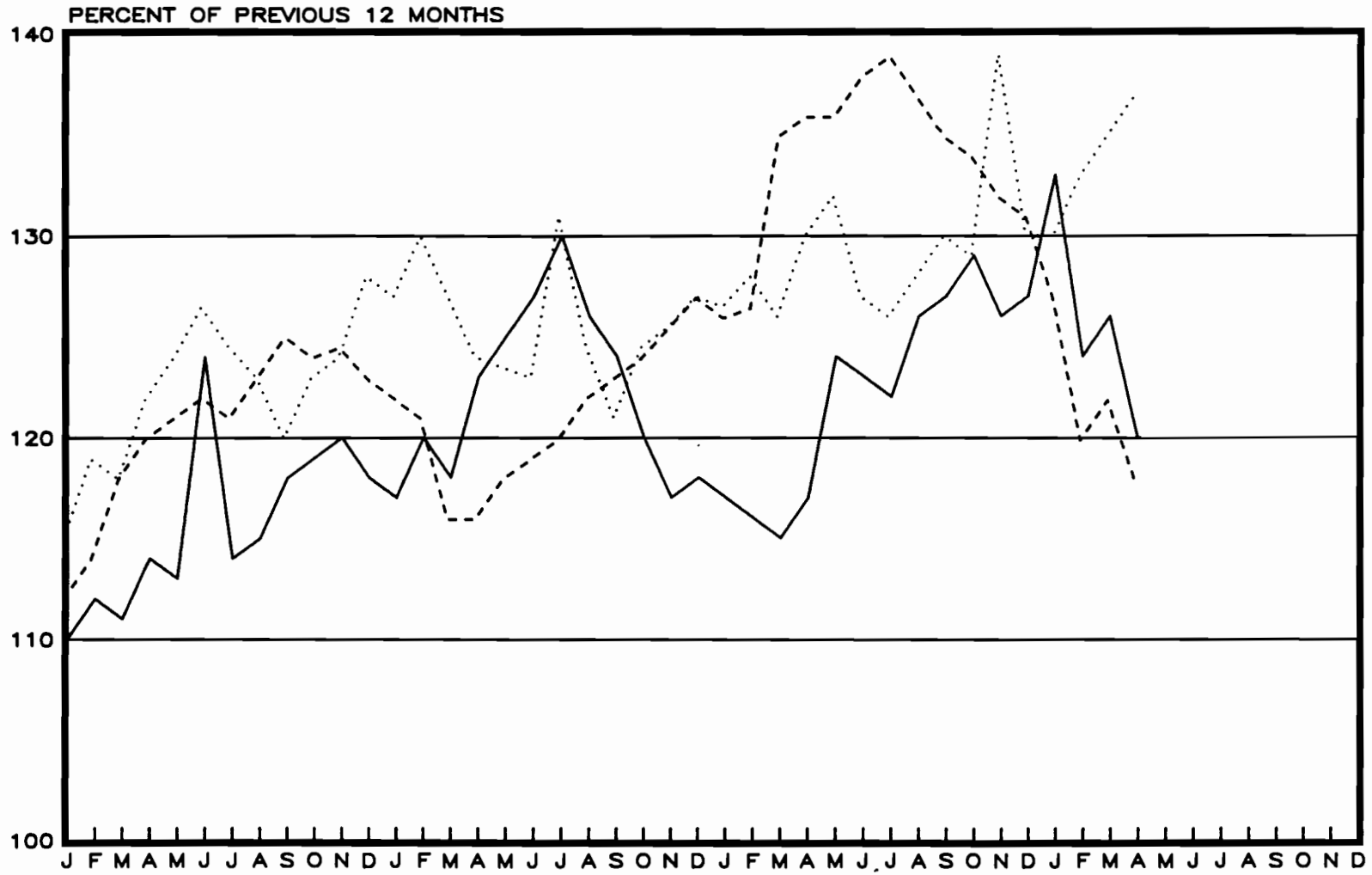
PERIOD EXPENSE CONTROL

12—MONTH MOVING AVERAGE

R&D EXPENSE HISTORY ——————

SALES HISTORY - - - - - - - -

ADMIN EXPENSE HISTORY .....

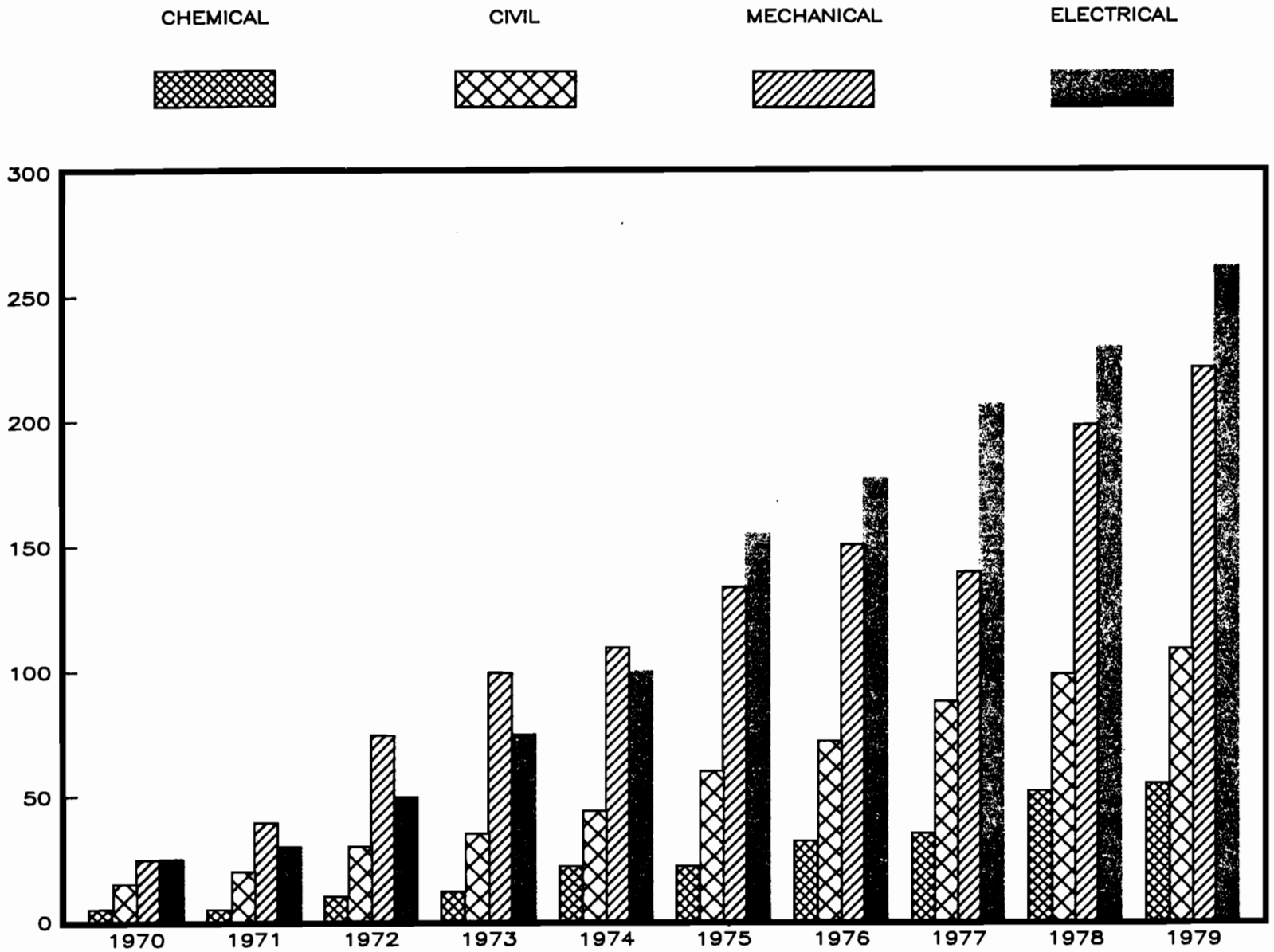


(1977—1980)

THIS IS A REPRODUCTION OF AN ORIGINAL PLOT CREATED ON AN HP7221 PLOTTER BY HP-DSG/3000 SOFTWARE

P/N: 5953—4075

ENGINEERING GRADUATES BY CATEGORY



THIS IS A REPRODUCTION OF AN ORIGINAL PLOT CREATED
ON AN HP7221 PLOTTER BY HP-DSG/3000 SOFTWARE

P/N: 5953-4077

30c

Want to do it Yourself ?

by Babs Brownyard, Information Systems Division

Hewlett Packard has recently introduced a new group of training products: the Self-Paced Learning Series for HP3000 users. These stand-alone packages enable students to proceed through instructional material at their own pace, studying any time and any place they have access to an HP3000. These new courses provide HP customers with a cost-effective means of training many end-users without interruption to their normal work schedules or travel to training centers.

The courses are designed for users who are new to the HP3000, introducing them to various facets of the system through illustrated texts, terminal interaction, and other media appropriate to the topic.

Three courses are currently available:

- HP3000: A Guided Tour--for the nonprogrammer, an introduction to the HP3000 and its wealth of information handling tools. This course is an ideal means for the new user to become comfortable interacting with the system. (22835A, $175)

- Learning COBOL II--an entry level course for COBOL programmers, it lays the fundamental groundwork needed prior to classroom training. (22832A, $450)

- Using DSG/3000--teaches non-programmers (managerial, administrative, and clerical) the procedure of chart design from conception to plotter output. (22833A, $250)

More information, as well as demonstrations of these courses, is available at your local sales office. They may be ordered through the sales office, and in the United States they may also be ordered from Computer Supplies Operation by telephone: 800 538-8787 toll free, or in California 408 738-4133 collect.

New RPG Utility - Extra Function Sort (XSORT)

by Kathy Weiler, Information Systems Division

Bundled with RPG/3000 version A.04.07 you will find a new utility "Extra Function Sort for RPG/3000", or XSORT. XSORT is a fully supported version of the previously unsupported RPG utility, SORT3, which has been available through the field organization for several years.

XSORT provides all the capabilities of SORT/3000, plus three additional functions which are required by RPG users in particular:

- Selection by "Include" and "Omit" specifications of subsets of the input records prior to sorting
- Use of different sort keys for each subset of input records
- Reformatting of data records on output after sorting

The primary purpose of XSORT is to facilitate conversion of sort procedures from IBM System/3 $DSORT and IBM System/32 and System/34 #GSORT. The syntax and semantics of XSORT specifications are identical to $DSORT and #GSORT specifications, with the following exceptions:

1) XSORT does NOT provide:

   - Support for data types "Z" and "D" for independent use of EBCDIC character "Zone" and "Digit" portions
   - Summary Sort capability (SORTRS)
   - The IBM #GSORT reserved "keywords" UDATE, UMONTH, UDAY, and UYEAR
   - Workfile Data Verification, as controlled by IBM $DSORT and #GSORT Header Column 34.
   - A "Null Output Message" option, as controlled by IBM #GSORT Header Column 36

2) XSORT DOES provide:

   - Support to handle up to 9 input files
   - A "Count-Only" Pass (SORTC) capability
   - Use of tape and other non-disc MPE files for input and output
   - An additional "special options" specification record (identified by an "S" in column 6) which provides additional global controls including:

- ADDROUT file first record number
- Output file existence check
- Maximum number of records to be sorted
- Diagnostic stack dump capabilities

XSORT formal file designators are XSORTIN, XSORTIN1,...XSORTIN9, XSORTOUT, and SPECCARD. The names SORT3IN, SORT3IN1,...SORT3IN9, and SORT3OUT are also allowed to provide upward compatibility for SORT3 users.

The RPG/3000 Utilities Reference Manual, part number 32104-90006, contains a full description of XSORT specifications and XSORT's use within the MPE operating environment.

HPIB 2613, 2617, and 2619 Line Printers

by Sue Thompson, Data Communications Operation

ANNOUNCEMENT!

HP 2613, 2617, and 2619 line printers have a new home. For the first time, the HP3000 Series 30/33/44 Systems can enjoy high speed line printers.

This is made possible by a new PCA board (HP26069A), and a new line printer driver (HIOLPRT2).

The HP26069A PCA translates HPIB protocol to differential protocol. You may order it as an option for new line printers or as a stand-alone for upgrading existing printers. All of the necessary hardware is available from the Boise Division.

The new driver is geared specifically toward chain printer control logic versus the dot matrix logic found in the 2608 printer. This software was first distributed as a product tape for the 2028 installation tape and is integrated in the 2052 IT.

Software Changes in IML/3000 Version A.00.01

by Connie Ishida, Data Communications Operation

Attention any user of the IML Intrinsics! Do you have a program which calls the intrinsic READSCREEN and then searches the array INTSCREEN for attribute characters, by checking for any high order bits set? If the answer is yes, then you should modify your program now. In the future, IML will be supporting larger than seven bit character codes, such as Katakana, so we will stop using the high order bit to designate an attribute character. The IML release A.00.01 still utilizes this high order bit, but it is advisable to modify your program now. To aid you in this change we have included a new intrinsic, ATTRLIST, in version A.00.01 of IML. This new intrinsic returns an array of the locations of attribute characters within any portion (or all) of the internal screen image. If you don't know the number of attributes within a particular portion of the screen area, and you provide an array which is too small to hold all of the locations of the attributes in that area, then ATTRLIST will fill your array and return a parameter which indicates the actual number of attributes in that screen area. If you deal with the entire screen, then you may call SCREENATTR first to determine the actual number of attribute characters (or fields) within the entire screen. The ATTRLIST intrinsic may prove very useful to other users as well as to the users of READSCREEN. The new edition of the IML Reference Manual (32229-90001) contains complete documentation on ATTRLIST in Chapter two.

Most IML users will find the second major change in revision A.00.01 very helpful. The IML manager DISPLAY command has been expanded to show which IML devices are in use and which are available for use, for every IML line configured in your system. The command previously only displayed the configuration file name for each IML line configured in the system. Now you can do a display to see which users, programs and acquired terminals are active. The manager can do a display to find an available device, acquire that device, and then do another display to verify that the device was indeed acquired. The command will also be useful to check the status of the IML line when the manager wishes to STOP or KILL IML. The new edition of the IML Reference Manual also contains complete documentation on this expanded command.

IML has now been verified on systems utilizing HP-IB type INP's. Therefore, IML is now supported on the HP3000 Series 30, 33 and 44. The new HP2626 terminal can be used with IDF. Please note that this terminal does not provide the half-bright display option. Another display option will have to be selected for use

with IDF.

With this release, IML also supports printer emulation through the Inquiry and Development Facility (IDF). An IBM 328X printer may be emulated using an HP2608A or 2631A on an HP-IB based system. The printers supported by IDF on Series II/III are HP2608A, 2613A, 2617A and 2619A. For further information on printer support consult the new edition of the IML Reference Manual. Support for printer emulation through the IML Intrinsics was already provided in the initial release of the product.

Customer experience with numeric lock has led us to remove IDF support for numeric lock and for the use of an asterisk (*) within a numeric field to denote a DUP key. On an IBM 3270, numeric lock is an option and, in fact, can be overridden in various manners depending upon the IBM keyboard. The customer problem arose from the fact that he had not purchased the numeric lock on his existing IBM terminals. His host application program had declared most of its fields to be numeric but, had then utilized alphanumeric data within these "numeric" fields. Since their IBM terminals did not have the numeric lock option, they accepted the data. The initial release of IML would not accept the alphanumeric data in these fields. As IDF cannot provide an option or an override, support for numeric lock has been removed in release A.00.01. Likewise, since IDF could only distinguish the asterisk as a DUP key within numeric fields and numeric-only is not being enforced, we also removed support from IDF for this indication of a DUP character. The impact on users of the IML Intrinsics is minimal, since you may still enter a DUP character code into STREAM3270 and WRITEFIELD. The user program can easily ensure that only numeric type data is transmitted to the host within numeric fields.

The first edition of the IML Reference Manual contained references to IDF terminals being used in "display only" mode. The cover sheet to the manual, however, indicated that this mode was not supported in the first release of the product. We have since determined that display only terminals as designed in IDF do not provide all of the features of the IBM keyboard-less terminals, and to provide this support would require changes to the user interface. Therefore IDF does not support display only terminals and references to this mode have been removed from the Reference Manual.

A large number of the outstanding software problems have been resolved in this first revision of IML/3000 software. The new edition of the IML Reference Manual reflects this and the changes described above; also, a number of typographical errors present in the first edition have been corrected.

A NEW COURSE:

The IML/3000 Training/Consulting course is now available. During the training portion of the course, you will receive instruction in the fundamentals of IML/3000 operation, managing IML on the HP 3000, programming with IML Intrinsics, and the use of the Inquiry and Development Facility. The course will consist of lectures and lab exercises that will be tailored to the needs of the your installation.

The consultation part of the course is intended to help you make implementation decisions that will help assure successful use of IML/3000.

C-MIT Brings Changes in Command for Tracing RJE/3000 Communications Lines

by Steve Stauss, Data Communications Operation

NOTE: THE FOLLOWING IS IMPORTANT TO ALL USERS WHO EMPLOY THE CS TRACE FACILITY TO TRACE THEIR RJE/3000 LINE(S). THIS INCLUDES USERS WHO INITIATE TRACE VIA A STREAM JOB, RJE/3000 COMMAND FILE, UDC, OR SOME OTHER AUTOMATED PROCEDURE!!!

The Syntax of the "TRACE" parameter of the RJE/3000 #RJLINE command has been revised for the C-MIT. This revision brings the syntax into conformity with the syntax of the TRACE command/parameter used in other subsystems. Also, the Trace mask has been expanded to eight bits (the two most significant bits of the mask byte are now used) and additional entry types are available.

Previously, the #RJLINE "TRACE" parameter had this syntax:

```
[;TRACE=[[ALL][,mask][,[numentries][,WRAP]]]]]
```

The syntax is now as follows:

```
          {,}
[;TRACE{=}ON[,[ALL][,[mask][,[numentries][,[WRAP]
    [,tracefilename]]]]]]
```

(Notice the addition of the "ON" and "tracefilename" subparameters. Also, note that a comma may now be used in place of the equal sign following the word "TRACE".)

The TRACE "sub-parameters" now have these definitions:

ALL -- If ALL is chosen, all line activity, not just errors, will be traced.

mask -- An octal number, preceded by a percent (%) symbol, that indicates the type of trace activity to record in the trace file. The user may select up to eight different categories of entry types; these may be chosen individually or in combination with others. Entry categories and their corresponding octal codes are:

```
%000 = Use default mask
%001 = PSTX entries
%002 = PSCT, PPOL, and PSEL entries
%004 = PRTX entries
%010 = PRCT entries
%020 = POPR and PEDT entries
%040 = PSTN entries
%100 = INP interconnect entries
%200 = IML control unit state transition
       entries

PCMP events are always recorded, no matter
what the mask.
```

To request a combination of entry categories, enter the sum of the codes of the individual entry categories desired. For example, consider the composition of the default mask, which is %37:

```
%001 (PSTX)
%002 (PSCT, PPOL, PSEL)
%004 (PRTX)
%010 (PRCT)
%020 (POPR, PEDT)
----
% 37 (PSTX, PSCT, PPOL, PSEL, PRTX, PRCT,
      POPR, PEDT)
```

Entry types are defined as follows:

PSTX: Send text. Created each time the driver sends a text message to the remote station. Contains a byte-for-byte copy of what was sent.

PSCT: Send Control Sequence. Generated each time the driver sends a control sequence to the remote station. Contains a byte-for-byte copy of what was sent.

PPOL: Send Polling Sequence. Generated each time driver sends a polling sequence. Contains a byte-for-byte copy of what was sent.

PSEL: Send Selection Sequence. Created each time driver sends a selection sequence. Contains byte-for-byte copy of what sent; all binary synchronous protocol characters are shown.

PRTX: Receive Text. Generated each time a text message is received from the remote station. Contains byte-for-byte copy of what was received.

PRCT: Receive Control Sequence. Generated each time control sequence received from the remote station. Byte-for-byte copy of what was received.

POPR: Operation. This entry is generated each time the physical driver is called upon to perform an operation. The entry tells what operation was requested.

PEDT: Editor. Generated each time a text message or control character sequence is received from the remote station. In case of text, contains the first words of the user's buffer; control characters, pad characters, and CRC characters are omitted. In the case of a control character sequence, the PEDT entry is a mnemonic phrase telling what was received.

PSTN: State Transition. Generated each time the driver shifts from one internal state to another. The entry shows what event has just completed and what action the driver will take in response to the event.

INP Interconnect: Avoid requesting a trace of interconnect events unless instructed to trace them by Hewlett-Packard. There are 18 different IC entry types, and if they are traced, trace file records can fill very quickly.

numentries -- The value supplied by the user will be used to determine the maximum number of entries to write to a trace record (i.e., it is used to determine the trace record size). The value input by the user should be an integer multiple of eight, not greater than 248 for a line with the SSLC or HSI or, for a line with the INP, not greater than 24. If the value supplied is not an integer multiple of eight, RJE/3000 will round it down so it complies with this requirement.

The default value of numentries is 24.

One trace record is written for each CS intrinsic call. When the "ALL" parameter is set, all activities associated with each intrinsic call can be written to a record (unless screen out by the mask).

Note: If the record is too small to hold all entries (numentries is too small), excess entries will be discarded or "wrapped" to the beginning of the record. Whether they are discarded or wrapped depends on whether the WRAP parameter was chosen.

WRAP -- By default, entries are written to a trace record in a linear fashion; once a record is full (numentries of entries have been written), additional events are not recorded. If WRAP is specified, entries will be written in a circular fashion; that is, once the record is full, the CS Trace facility will return to beginning of the record and continue writing.

Note: This parameter does not apply to a trace file. If the trace file becomes full, it is automatically cleared and writing to it continues. Thus, the trace file can expand and contract over time.

tracefilename -- The name the user wants the trace file to have. The Trace facility will save the trace file (which it builds) with this name. If no trace file name is supplied, a default name will be used. The default will be

RJETRCEn, RJETRCnn, or RJETRnnn

with n, nn, or nnn set to the logical device number of the line.

Editor's Note: This article is a reprint of an article which was initially published in the proceedings of the "North American General Systems Users Group Meeting" held in San Jose, California.

Debugging COBOL II/3000 Programs

by Greg Gloss, Information Systems Division

This paper provides some tips for debugging programs compiled with the new COBOL II/3000 compiler. A discussion of compile time aids is followed by a description of the run time environment and guidance for using MPE DEBUG to debug your programs interactively.

COMPILE TIME AIDS

Users of COBOL/3000 will notice three major changes in compile time debugging aids with COBOL II/3000. The first change is that diagnostic messages are now listed together at the end of the listing. In the following sample program the hyphen was left out of N-2 in line 22. Note that the diagnostic listing references both a compiler generated line number (5 digits) and the 6 digit sequence field.

```
PAGE 0001   HEWLETT-PACKARD 32233A.00.00  COBOL II/3000

00001          001000$CONTROL CROSSREF,VERBS,MAP
00002          001100 IDENTIFICATION DIVISION.
00003          001200 PROGRAM-ID. SAMPLE.
00004          001300 ENVIRONMENT DIVISION.
00005          001400 DATA DIVISION.
00006          001500 WORKING-STORAGE SECTION.
00007          001600 01  GRP.
00008          001700     05  N-1 PIC S9(4) VALUE 0
00009          001800            SIGN IS LEADING SEPARATE.
00010          001900     05  N-2 PIC S9(4) VALUE 0
00011          002000            SIGN IS LEADING SEPARATE.
00012          002100     05  N-3 PIC S9(4) COMP-3.
00013          002200 PROCEDURE DIVISION.
00014          002300 SEC-1 SECTION 01.
00015          002400 BEGIN-HERE.
00016          002500     DISPLAY "ENTER N-1, 4 DIGITS".
00017          002600     ACCEPT N-1 FREE.
00018          002700     DISPLAY "N-1 = ", N-1.
00019          002800 SEC-2 SECTION 02.
00020          002900 PAR-2.
00021          003000     DISPLAY "ENTER N-2, 4 DIGITS".
```

```
00022           003100      ACCEPT N2 FREE.
00023           003200      DISPLAY "N-2 = ", N-2.
00024           003300      MOVE N-2 TO N-3.
00025           003400 SEC-1A SECTION 01.
00026           003500 STOP-HERE.
00027           003600      STOP RUN.


COBOL ERRORS:

 LINE #  SEQ #   COL  ERROR  SVRTY  TEXT OF MESSAGE
-------------------------------------------------------------

 00022  003100   21    356     Q     UNDEFINED DATA NAME N2

0 ERRORS, 1 QUESTIONABLE,  0 WARNINGS
```

The other new features which have been added are a cross reference capability and a verb map listing. The CROSSREF option on the $CONTROL command generates an alphabetical listing of user-defined symbols in the program with the line numbers in which they are referenced. The following listing shows the cross reference generated after the above program was corrected.

```
COBOL CROSS REFERENCE LISTING


BEGIN-HERE
00015

GRP
00007

N-1
00008    00017    00018

N-2
00010    00022    00023    00024

N-3
00012    00024

PAR-2
00020

SEC-1
00014

SEC-1A
00025

SEC-2
00019

STOP-HERE
00026
```

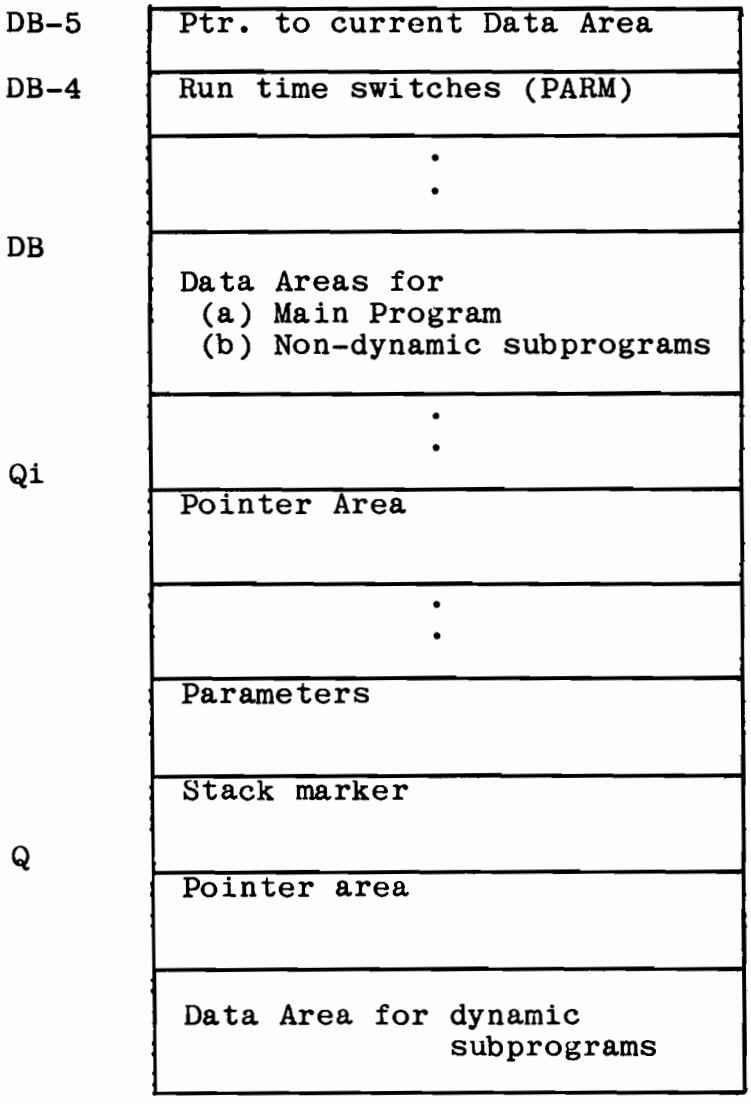
For example, N-2 is referenced in 4 places: lines 10, 22, 23, and 24.
The third change is the capability to list the starting address of code generated for each verb instead of just each paragraph. The VERBS option of the $CONTROL command lists these addresses which are relative to the starting address of the RBM (Relocatable Binary Module) as shown in the PMAP. The following listing shows the address of the ACCEPT statement in line 17 to be 33 relative to the start of the RBM SEC101'. This address can be used with MPE DEBUG to set a breakpoint at the start of this statement.

```
 LVL SOURCE NAME                    BASE DISPL    SIZE    USAGE


 WORKING-STORAGE SECTION

 01  GRP                           Q+2: 000350   000015 DISP
 05   N-1                           Q+2: 000350   000005 DISP
 05   N-2                           Q+2: 000355   000005 DISP
 05   N-3                           Q+2: 000362   000003 COMP-3

PAGE 0003/COBTEXT  SAMPLE           SYMBOL TABLE MAP
 LINE #    PB-LOC        PROCEDURE NAME/VERB       INTERNAL

 00014     000003      SEC-1                       SEC101'
 00015     000003          BEGIN-HERE
 00016     000003              DISPLAY
 00017     000033              ACCEPT
 00018     000042              DISPLAY
 00019     000003      SEC-2                       SEC202'
 00020     000003          PAR-2
 00021     000003              DISPLAY
 00022     000033              ACCEPT
 00023     000042              DISPLAY
 00024     000071              MOVE
 00025     000003      SEC-1A                      SEC1A01'
 00026     000003          STOP-HERE
 00027     000003              STOP
```

RUN TIME ENVIRONMENT

[NOTE: All information presented in this area is for help in debugging programs. The run time structure is subject to change in future releases so applications should not be developed which depend on this structure remaining constant.]

The stack layout for a COBOL II program is shown below. Each program unit (Main program or subprogram) has two areas in the data stack: a data area and a pointer area. For main programs and non-dynamic subprograms, the data areas are located between DB and Q initial. The pointer areas for all types of program units are dynamically allocated starting with Q+1 upon entry to the program unit. For dynamic subprograms, the data area is allocated after the pointer area.



Pointer Area

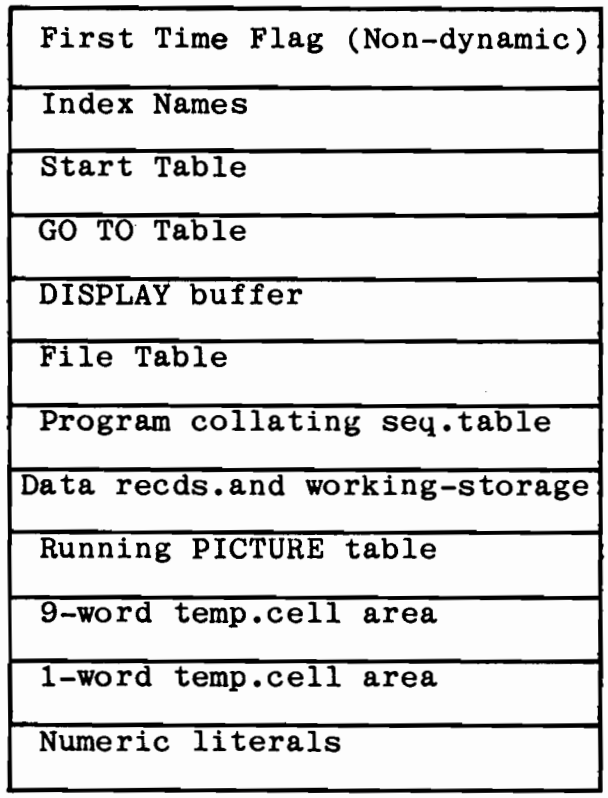
The pointer area contains the addresses of data structures in the data area and other information local to the program unit. All types of COBOL II program units have the same format for the pointer area except that main programs do not have an area for Linkage Section addresses. For subprograms, each 01 or 77 level item in the Linkage Section is assigned a 1-word location at or above Q+13 for the word address. This address is moved from the appropriate Q-negative location upon entry to the subprogram. In addition, another 1-word location is allocated for the corresponding byte address. If a Linkage Section item is not referenced in the USING phrase of the Procedure Division header or ENTRY statement, these locations will contain an illegal address. The format of the pointer area is shown below:

Q+1	word address of data area
Q+2	byte address of data area
Q+3	DECIMAL PT. : COMMA
Q+4	:SE F :#params.: CURRENCY SIGN :
Q+5	byte address of 9-word temps.
Q+6	word address of 1-word temps.
Q+7	byte pointer of numeric lits.
Q+8	SORT-MERGE switch
Q+9	Start Table address
Q+10	File Table address
Q+11	previous value in DB-5
Q+12	reserved
Q+13	word addresses of 01 and 77 items in LINKAGE SECTION (N)
Q+13+N	byte addresses of 01 and 77 items in LINKAGE SECTION

DECIMAL PT. and COMMA are character representations of DECIMAL-POINT and COMMA.
F(1 bit)=First-time flag for PERFORM statements
SE(1 bit)= SIZE ERROR flag: 1- size error occurred
#params(6 bits)= no.of parameters
CURRENCY SIGN = character representing currency sign

Data Area

The data area contains a combination of compiler generated structures and user defined data areas. These areas are described below:



1. First Time Flag (Non-dynamic Subprograms only).
   A 1-word value indicating whether or not it is the first call to a Non-dynamic subprogram.

2. Index Names (Optional)
   One word is allocated for the value of each index-name declared in the program.

3. Start Table.
   Each paragraph/section generates a two-word entry containing its starting PB-address and segment number.

4. GO TO Table (Optional).
   Each Alterable GO TO (a paragraph whose first or only statement is a simple GO TO) generates a three word entry for use with the ALTER statement.

5. DISPLAY Buffer (Optional)
   A 200-byte buffer for values to be DISPLAYed.

6. File Table (Optional).
   Contains information pertaining to each user specfied file.

7. Program collating seq. Table (Optional).
   A table containing the mapping for characters if a PROGRAM COLLATING SEQUENCE is specified in the Environment Division.

8. Data Records and Working Storage (Optional).
   This area contains the values of User defined data items declared in the File and Working-Storage Sections.

9. Running Picture Table (Optional).
   Contains information pertaining to data items and is used by the run-time library for certain constructs.

10. 9-word temp cell area (Optional).
    Used for intermediate values in arithmetic computations.

11. 1-word temp cell area (Optional).
    Used for intermediate values in arithmetic computations.

12. Numeric Literals (Optional).
    Contains representations of some of the numeric literals used by the program.

Code Segmentation

Each COBOL II program unit generates at least two RBMs. The first RBM is an initialization module. For main programs, this unit is the Outer Block where control is transferred when the program is started. The name of this module is the same as the PROGRAM-ID name (stripped of hyphens). For subprograms, an apostrophe is appended to the PROGRAM-ID name to form the name of the initialization module. The Procedure Division will generate one or more RBMs depending on the segment numbers (if any) which are specified in the section headers. If no segment numbers are specified, the entire procedure division goes in one RBM. Consecutive sections with the same segment number are placed in the same RBM. The RBM name is listed in the compile-time MAP under the heading "INTERNAL NAME." The initialization RBM is put into the same code segment as the last RBM of the Procedure Division. Non-contiguous sections with the same segment number will

generate multiple RBMs but they all will be put into the same code segment.

The PMAP from the sample program is shown below:

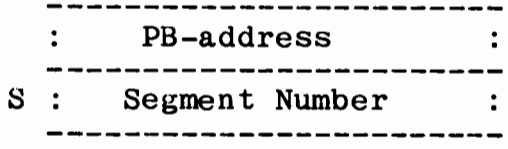
```
SEC202'            0
   NAME             STT  CODE ENTRY SEG
   SEC202'            1     0     0
   C'DISPLAY          2                 ?
   C'DISPLAY'FIN      3                 ?
   C'DISPLAY'INIT     4                 ?
   C'DISPLAY'L        5                 ?
   ACCEPT'FREE'C      6                 ?
   SEGMENT LENGTH          124
SEC101'            1
   NAME             STT  CODE ENTRY SEG
   SEC1A01'           1     0     0
   TERMINATE'         4                 ?
   QUIT               5                 ?
   SAMPLE             2     6     6
   DEBUG              6                 ?
   COBOLTRAP          7                 ?
   SEC202'           10                 0
   SEC101'            3   216   216
   C'DISPLAY         11                 ?
   C'DISPLAY'FIN     12                 ?
   C'DISPLAY'INIT    13                 ?
   C'DISPLAY'L       14                 ?
   ACCEPT'FREE'C     15                 ?
   SEGMENT LENGTH          334
```

Notice that sections SEC-1 and SEC-1A generate different RBMs (SEC101' and SEC1A01'), but are both in the same code segment because they both have a segment number of 01. Since SEC-1A is the last section of the Procedure Division, the initialization module SAMPLE is also put in this segment. If the program were changed such that SEC-1A came between SEC-1 and SEC-2 then both SEC-1 and SEC-1A would be in the same RBM and the initialization module would go in the same segment as SEC-2.

Implementation of PERFORM and GO TO

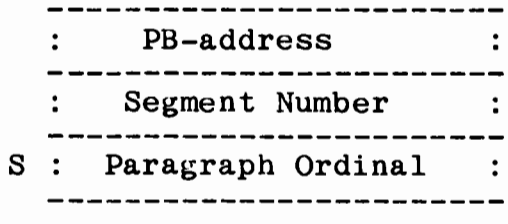
The implementation of PERFORM and GO TO statements has been changed with COBOL II/3000. Three new instructions have been added to the machine instruction set to perform these operations. For GO TO statements across section boundaries, the new External Branch (XBR) instruction is used as follows:

The two word value from the Start Table for the paragraph being branched to is loaded onto the top of the stack

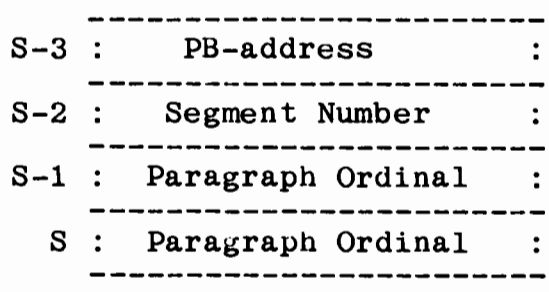


and the XBR instruction transfers control to the specified location, deleting the 2 words from the stack.

For all PERFORM statements, the Paragraph Call (PARC) instruction is used to initiate a PERFORM. First, the two word Start Table entry for the starting paragraph of the PERFORM is loaded onto the top of the stack followed by the ordinal of the last paragraph being performed.



The PARC instruction replaces the PB address and Segment Number with the appropriate return values and then transfers control. To return from a PERFORM statement, the End Paragraph (ENDP) instruction is used. First the ordinal of the paragraph being concluded is loaded onto the stack.



The ENDP instruction compares the two paragraph ordinals and if equal returns control to the location specified in S-2 and S-3 and deletes the four words from the stack. If the paragraph ordinals are not equal only one word is deleted from the stack and control falls through to the next paragraph.

USING MPE DEBUG

To use MPE DEBUG you should have a compilation listing of the program units with the MAP and VERBS options specified on the $CONTROL command together with the PMAP for the program file. The CROSSREF option on the compilation listing may also prove useful. There are two ways to get into DEBUG with a COBOL II program. The first is to specify DEBUG on the :PREPRUN or :RUN command. The second is to specify the DEBUG option on the $CONTROL command when the program is compiled. This option sets the CONTROL-Y trap so that pressing Control-Y when the program is executing will invoke DEBUG.

To debug the sample program shown earlier,

```
:RUN SAMPPROG,DEBUG
```

To set a breakpoint, get the offset from the VERBMAP for the statement and add the starting address of the code module as shown in the PMAP. For example, to set a breakpoint at the ACCEPT statement in line 17, get the logical segment number of the segment containing SEC101' (which is 1) and the starting address of the module (which is 216). From the VERBMAP, the offset for the ACCEPT statement is 33. So, when the program starts,

```
*DEBUG* 1.6
?B1.216+33
```

sets a breakpoint at the start of the first ACCEPT statement. To set another breakpoint at the STOP RUN statement in line 27, you will notice that module SEC1A01' starts at PB+0 and the STOP RUN statement starts at 3.

```
?B1.3
?R
```

The R will resume execution of the program.

```
ENTER N-1, 4 DIGITS

*BREAK* 1.251
?
```

Now we are at the start of the ACCEPT statement. To verify the initial value of N-1 as 0, display its contents.

```
?D ('Q+2'+350)/2,3,A
DB+164    +0000+
```

The 'Q+2' means the contents of Q+2 which contains the byte address of the data area. The 350 is the byte address of N-1 as shown in the MAP. The /2 converts the byte address to a word address and the 3 specifies the length as 3 words. Since the item is USAGE DISPLAY, the A specifies to display the value in ASCII.

Since the item is actually only 5 bytes long, displaying 3 words picks up the first byte of the next item also. For items in the Linkage Section the Q+2 would have to be replaced by the value shown in the BASE column of the MAP. This value is currently displayed as a positive decimal value (Q14 means Q+14 (decimal)). To use this value in a DEBUG command, precede the numeric offset by #; for example, Q+#14. A future release (A.00.02) will change this to an octal representation to ease using DEBUG.

To continue running until the other breakpoint, the following is keyed in:

```
?R

123                    Value of N-1 keyed in
N-1 = +0123
ENTER N-2, 4 DIGITS
-1                     Value of N-2 keyed in
N-2 = -0001

*BREAK* 1.3
?
```

Now we are at the STOP RUN statement. To verify that N-3 did get the proper value from the MOVE statement,

```
?D ('Q+2'+362)/2,2,H
DB+171    0000 1D00
```

Since N-3 is a COMP-3 item, the H (for Hexadecimal) representation was displayed. The value of N-3 is shown as 00001D (The D being the representation of minus).

Using the same information sources (the MAP, VERBMAP, and PMAP), values of data items may also be changed in the middle of executing a program.

Undocumented Commands of the Segmenter

by Doug Pollard, Computer Systems Division

This article provides more detail on four commands and an intrinsic which appear in the MPE Software Pocket Guide but not in the Segmenter Reference Manual. It also describes an enhancement to the LISTSL and LISTUSL commands.

The names of the commands are CLEANSL, CLEANUSL, COPYSL, and COPYUSL. CLEANUSL is the name of the intrinsic.

In the examples of this article, commands a user would enter are in lower case type, and comments by the author are set off by double "greater than" and "less than" symbols.

CLEANSL
(Segmenter Command)

Copies the contents of the currently managed Segmented Library file to a new SL file, eliminating any fragments of free space in each of the segments.

Syntax

```
-CLEANSL [filename]
```

Parameter

filename	The name of the new SL file. If you omit this parameter, the Segmenter will purge the old SL file, giving its name to the new file.

Operation

The new SL file will be of the same size as the old one; that is, it may hold up to the same maximum number of records.

The new SL file becomes the currently managed SL.

Example

```
-sl sl                    << The Segmenter copies the contents
-cleansl newsl               of the file SL into the new file
                             NEWSL.  Note that NEWSL is now the
                             currently managed SL.>>
```

COPYSL
(Segmenter Command)

Copies the contents of the currently managed Segmented Library file to a new SL file, eliminating any fragments of free space while giving you the ability to control the size of the new file.

Syntax

```
-COPYSL percent [,filename]
```

Parameters

percent	The amount of extra file space you want the new file to have, expressed as a percentage of the minimum amount of space needed to hold the segments. This parameter must be an integer, not less than 0 or greater than 9900.
filename	The name of the new SL file. If you omit this parameter, the Segmenter will purge the old SL file, giving its name to the new file.

Operation

This command is different from CLEANSL in that you may change the amount of file space used by the file.

The new SL becomes the currently managed SL file.

Example

```
-sl sl                    << The Segmenter copies the contents
-copysl 40, newsl            of SL into the new file NEWSL,
                             building NEWSL with forty percent
                             more free space than the minimum
                             it needs to hold the segments. >>
```

CLEANUSL
(Segmenter Command)

Copies only the active Relocatable Binary Modules of the currently managed User Subprogram Library to a new USL file.

Syntax

```
-CLEANUSL [filename]
```

Parameter

filename	Is the name of the new USL file. If you omit this parameter, the Segmenter will purge the old file, giving its name to the new USL.

Operation

The new USL will be the same size as the old one, that is, it will hold up to the same maximum number of records.

In response to this command, the Segmenter calls the CLEANUSL intrinsic.

The new USL file becomes the currently managed USL.

Example

```
-usl searusl                  << The user first identifies
-listusl                        SEARUSL as the currently
                                managed USL. >>

USL FILE SEARUSL.SEGMENT.SUB3000

SEARCHSEG
   SEARCHLINE          27  P  A C N R
   SEARCHLINE          27  P  I C N R
ASKSEG
   ASKFORMAT           15  P  I C N R <<SEARUSL contains 4
SEG'                                      inactive RBM's among
   WRITENUMSONLY       16  P  I C N R   its 3 segments. >>
   WRITENUMSONLY       16  P  I C N R
   OB'                255  OB A C N
   WRITENUMSONLY       16  P  A C N R
   ASKCHAR             17  P  A C N R
   ASKNAME             13  P  A C N R
```

```
FILE SIZE    2000(    10.  0)
DIR. USED     563(     2.163)   INFO USED    1132(  4    .132)
DIR. GARB.      0(     0.  0)   INFO GARB.      0(  0    .  0)
DIR. AVAIL.    15(     0. 15)   INFO AVAIL.    46(  0    . 46)

-cleanusl searusl7

-listusl

USL FILE SEARUSL7.SEGMENT.SUB3000

SEARCHSEG                                  << SEARUSL7, the new file,
   SEARCHLINE          27  P  A C N R         contains only the active
ASKSEG                                        RBM's of SEARUSL. >>
SEG'
   OB'                255  OB A C N    << The segment ASKSEG, which
   WRITENUMSONLY       16  P  A C N R      contained no active RBM's
   ASKCHAR             17  P  A C N R      in SEARUSL, is still
   ASKNAME             13  P  A C N R      present. >>

FILE SIZE    2000(    10.  0)
DIR. USED     435(     2. 35)   INFO USED    1016(  4. 16)
DIR. GARB.      0(     0.  0)   INFO GARB.      0(  0.  0)
DIR. AVAIL.   143(     0.143)   INFO AVAIL.   162(  0.162)

   << Note that SEARUSL7, the new USL file, has
      more free space than SEARUSL. >>
```

COPYUSL
(Segmenter Command)

Copies all of the currently managed User Subprogram Library to a new USL file, while giving you the ability to control the size of the new USL.

Syntax

```
-COPYUSL  percent  [,filename]
```

Parameters

percent	The amount of free space you wish the new USL to have, expressed as a percentage of the minimum space needed to hold the file. This parameter must be an integer not less than zero or greater than 9900.

filename — The name of the new USL file. If you omit this parameter, the Segmenter will purge the current USL file, giving its name to the new USL.

Operation

Unlike the COPYSL command, the COPYUSL command does no "cleaning". This command will copy inactive Relocatable Binary Modules to the new file as well as active RBM's.

The new USL file becomes the currently managed USL.

Example

```
-usl searusl          << User designates SEARUSL as the currently
                         managed USL file. >>


-copyusl 100,searusl5    <<Segmenter copies all of SEARUSL
                           to new file SEARUSL5, giving
                           SEARUSL5 two times the minimum
                           amount of space needed to hold
                           its present contents. >>


-copyusl 0,searusl6      << Segmenter copies all of SEARUSL5
                            (now the currently managed USL
                            file) into the new file SEARUSL6,
                            giving SEARUSL6 no more than the
                            minimum amount of space needed to
                            hold its present contents. >>

-listusl                 << SEARUSL6 is now the currently
                            managed USL file. >>

USL FILE SEARUSL6.SEGMENT.SUB3000

                    .
                    .
                    .

FILE SIZE     2000(    10.  0)
DIR. USED      563(     2.163)   INFO USED     1132(  4.132)
DIR. GARB.       0(     0.  0)   INFO GARB.       0(  0.  0)
DIR. AVAIL.     15(     0. 15)   INFO AVAIL.     46(    . 46)
```

```
-usl searusl5
-listusl

USL FILE SEARUSL5.SEGMENT.SUB3000

                    .
                    .
                    .

FILE SIZE     3600(    17.  0)
DIR. USED      563(     2.163)    INFO USED     1132(  4.132)
DIR. GARB.       0(     0.  0)    INFO GARB.       0(  0.  0)
DIR. AVAIL.    415(     2. 15)    INFO AVAIL.   1246(  5. 46)

 << Note the difference in the amount of free space each
    file has. >>
```

```
LISTSL [segment_name]
(Enhancement to LISTSL command)

Will list only the segment you specify rather than all the seg-
ments in the currently managed Segmented Library.
```

Syntax

```
   -listsl [segment_name]
```

Parameter

segment_name	The name of the segment you would like to see listed. If you omit this parameter, the Segmenter will list all the segments in the currently managed SL file.

Example

```
-sl sl       << SL is the currently managed SL file.
-listsl         The Segmenter will list information on all
                of the segments in SL. >>

SL FILE SL.SEGMENT.SUB3000

SEGMENT    0 SEARCHSEG        LENGTH     34

   ENTRY POINTS     CHECK CAL STT  ADR
   SEARCHLINE         0    C    1     0
```

```
   EXTERNALS        CHECK STT SEG

100

SEGMENT    1 WRITESEG          LENGTH     20

   ENTRY POINTS     CHECK CAL STT  ADR
   WRITENUMSONLY      0    C    1     0

   EXTERNALS        CHECK STT SEG

010

USED          2400(    12.  0)   AVAILABLE   57400(  276.  0)

-listsl searchseg    << The Segmenter will list only
                        information on the segment
                        SEARCHSEG of the SL file. >>

SL FILE SL.SEGMENT.SUB3000

SEGMENT    0 SEARCHSEG         LENGTH     34

   ENTRY POINTS     CHECK CAL STT  ADR
   SEARCHLINE         0    C    1     0

   EXTERNALS        CHECK STT SEG

100
```

LISTUSL [segment_name]
(Enhancement to LISTUSL command)

Will list only the segment you specify rather than all the segments in the currently managed User Subprogram Library.

Syntax

```
-listusl [segment_name]
```

Parameter

segment_name	The name of the segment you would like to see listed. If you omit this parameter, the Segmenter will list all of the segments in the currently managed USL file.

Example

```
-usl searusl
-listusl

USL FILE SEARUSL.SEGMENT.SUB3000

WRITESEG
   WRITENUMSONLY       16  P  A C N R
SEARCHSEG
   SEARCHLINE          27  P  A C N R
   SEARCHLINE          27  P  I C N R
ASKSEG
   ASKFORMAT           15  P  I C N R
SEG'
   WRITENUMSONLY       16  P  I C N R
   WRITENUMSONLY       16  P  I C N R
   OB'                255  OB A C N
   ASKCHAR             17  P  A C N R
   ASKNAME             13  P  A C N R

FILE SIZE    2000(   10.  0)
DIR. USED     574(    2.174)   INFO USED    1132(  4.132)
DIR. GARB.      0(    0.  0)   INFO GARB.      0(  0.  0)
DIR. AVAIL.     4(    0.  4)   INFO AVAIL.    46(  0. 46)

-listusl searchseg    << Because the user specified the
                         segment name, the Segmenter will
                         list only the information on the
                         segment SEARCHSEG. >>

SEARCHSEG
   SEARCHLINE          27  P  A C N R
   SEARCHLINE          27  P  I C N R
```

CLEANUSL
(Segmenter Intrinsic)

Copies only the active entries of the currently managed User Subprogram Library file into a new, "clean," USL file.

Syntax

```
   I                    IV        BA
filenum := CLEANUSL (uslfnum,filename);
```

Parameters

uslfnum	Integer by value (required). A word identifier supplying the file number of the file.
filename	Byte array (required). The name to be given to the new USL file. It may be fully qualified.

Operation

If no error occurs, this intrinsic returns the file number of the new USL. In the event of an error, however, the integer uslfnum will contain the error number rather than the file number, and, unless your program checks the condition code, the results are unpredictable.

This intrinsic requires 2900 words of user stack space.

CLEANUSL creates the new USL file as a job temporary file. To make the file permanent you must call the FCLOSE intrinsic.

Condition Codes

CCE	Request granted. CLEANUSL returns the new file number in the integer uslfnum.
CCG	CLEANUSL does not return CCG.
CCL	Request denied. CLEANUSL returns one of the following error numbers in uslfnum:

Error Number	Meaning
0	Unexpected end of file marker on either the old or the new USL file.
1	Unexpected I/O error on either the old or the new USL file.
7	Unable to open new USL file.
12	Invalid USL file.

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.

New Computer Documentation Index

An extremely useful reference is now available from Hewlett-Packard. The Spring 1981 Computer Documentation Index 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 94088

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 pub-

lished. 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.

NEW MANUALS

Series 44 Console Operator's Guide
part number 30090-90013
January, 1981

A complete reference guide for the console operator of the exciting new HP 3000 Series 44 Computer System, this manual includes sections on system controls and the Control and Maintenance Processor, console operator commands, an operational overview, and much more. It is written in an easy-to-use reference format which refers the reader to additional discussion in other sections and other HP manuals.

Extra Function Sort for RPG/3000
(XSORT)
part number 32104-90006

XSORT is the fully supported version of the previously unsupported RPG utility, SORT3. Besides the capabilities of SORT3, XSORT offers three new features: 1) selection by records prior to sorting; 2) use of different sort keys for each subset of the input records; and 3) reformatting of data records on output after sorting.

The primary purpose of XSORT is to facilitate conversion of sort procedures from IBM System/3 $DSORT and System/32 and System/34 GSORT. The syntax and semantics of XSORT specifications are identical to $DSORT and GSORT with a few exceptions, the most obvious of which is XSORT's lack of support for independent zone and digit ("Z" and "D") portions of characters and the lack of a summary sort capability. XSORT can handle multiple file inputs, a "count only" pass, and an additional specification record--the Special Option--which provides more global controls.

The manual is intended to serve the needs of users with varying experience. Overviews of specifications are included for those who are familiar with similar programs. Detailed discussions of column entries and sample jobs provide less experienced users with information and guidance in developing specifications for their particular needs.

DSG/3000 Manual
part number 32250-90001
November, 1980

This manual documents HP Decision Support Graphics/3000, a new HP data display system that allows users to design, produce, and save business graphs drawn from information kept in a data file.

The manual explains how to use DSG/3000 interactively through a series of screen menus, or programmatically through a set of procedures.

The manual contains an alphabetized listing of the screen menus with definitions of individual fields for the interactive user. For programmatic users, the manual contains reference specifications for DSG/3000 procedures.

DSG/3000 Guide
part number 32250-90002
November, 1980

This small guide, spiral bound for easy use, briefly discusses such topics as filling in menus, correcting errors, and building data files. The guide also outlines how to design bar, line and pie charts and how to modify them. Charts of available textures and data point markers as well as a GRAPH roadmap are included.

NEW EDITIONS

FCOPY Reference Manual
part number 03000-90064
July, 1980

This new edition of the FCOPY Reference Manual has been enhanced by greater coverage of the FCOPY Subsystem. Also, several changes and additions have been made to the manual.

MPE Commands Reference Manual 4th Edition
part number 30000-90009
January, 1981

This new edition specifies commands and describes the MPE Command Interpreter as compatible with the MPE IV operating system. One of the major changes in this edition is the deletion of text discussions of the various commands which results in a more reference-type document. When additional discussion is included in other documents, the reader is referred to the appropriate manual. The :FILE command, in particular, has been updated to reflect many changes and corrections associated with MPE IV.

MPE Intrinsics Reference Manual
part number 30000-90010
January 1981

This new edition of the Intrinsics manual has been enhanced by further discussion on many of the existing intrinsics, as well as the addition of two new intrinsics, FREADBACKWARDS, and CREATEPROCESS. There is also a new section on InterProcess Communication and Circular Files.

System.Manager/System Supervisor Reference Manual
part number 30000-90014
February 1981

This edition reflects significant changes for the MPE IV version of the HP 3000 operating system, including the TUNE command for on-line tuning of the system, changes to the Initiator/Sysdump dialogues, new configuration data, and other changes relating to the new Series 44 system.

MPE Software Pocket Guide 7th Edition
part number 30000-90049
January, 1981

This new edition to the Pocket Guide encompasses the enhancements and changes to the operating system and subsystems that accompany MPE IV. Corrections and clarifications are also included.

Communications Handbook
part number 30000-90105
February, 1981

This new edition of the Communications Handbook, formerly the Data Communications Handbook, reflects many revisions and changes. Among other topics, the manual now covers IML/3000.

HP 30010A/30020A Intelligent Network Processor
Diagnostic Procedures Manual
part number 30010-90002
February, 1981

This new edition of the INP Diagnostic Procedures Manual includes the expanded tests that are now part of the Online Diagnostic/Support Monitor (DSM).

Series 30/33/44 Diagnostic Manual Set
part number 30070-60068
January, 1981

This set of manuals, which includes several new manuals and updates as well as new editions, covers the diagnostics for the Series 30, 33 and 44 computers.

Series 30/33 Console Operator's Guide
part number 30070-90025
December 1980

This new edition of the Console Operator's Guide includes changes made to the system by the latest version of the operating system, MPE IV. There are three new communications commands, an Overview of User Logging, and a new error message and recovery section.

DS/3000 Reference Manual
part number 32190-90001
February, 1981

This new edition of the DS/3000 Reference Manual contains many changes and additions associated with MPE IV. Among the topics covered is NFT, Network File Transfer.

MTS/3000 Multileaving Terminal Software
Reference Manual
part number 32193-90002
November, 1980

This new edition introduces the support of MTS/3000 on the Intelligent Network Processor, Series 30/33/44 computer systems, and the HP 262x family of multipoint display stations. In addition, Section II has been substantially revised, and material on the preparation of multipoint display stations for MTS/3000 operation has been gathered together into a separate appendix, Appendix C.

HP V/3000 Reference Manual 3rd Edition
part number 32209-90001
February, 1981

The third edition of the data entry and forms management system V/3000 Reference Manual documents the following enhancements:

Three new intrinsics which allow programmatic setting of labels for function keys

Support of the HP 2624A terminal including local editing

Support of the HP 2626 terminal including windows and workspaces

Support of the HP 3075, 3076 Data Capture terminals

Security display enhancement

Modified data tag feature of the HP 2624A terminal

The new edition also includes a reprinted listing of the SPL entry program showing usage of the new intrinsics.

IML/3000 Interactive Mainframe Link
Reference Manual
part number 32229-90001
November, 1980

This new edition documents several changes in the IML/3000 software and in the hardware that will run with IML. Also, various typographical errors that were present in the first edition have been corrected.

Operation of IML/3000 on the Series 30/33/44 computer systems and on the HP 2626A terminal has been verified.

Software changes include a new intrinsic, ATTRLIST, which returns the locations of attributes within any subsection of the screen, an expanded IML manager DISPLAY command, and elimination of IDF support of NUMERIC LOCK and the DUP key. All references to emulation of keyboardless terminals have been deleted from the manual, as IDF does not support all of the features of IBM keyboardless terminals. Suggested names for intrinsic parameters have been changed to provide for more consistency and greater ease of use.

MPE Debug/Stack Dump Update #2
Reference Manual
part number 30000-90012
October, 1980

This update contains error corrections as well as changes associated with MPE IV. Enhancements to Debug commands include:

Conditional breaking. This is available with the B and R commands.

Operations on processes other than your own. This is available with the B, R, and C commands.

Display of memory locations in actual code or hexadecimal characters. This is available with the D, M, and = commands.

Series III Console Operator's Guide
part number 30000-90013
December 1980

This update reflects changes made for MPEIV including three new communications commands. A new error message and recovery section has also been added, as well as an overview of User Logging.

MPE System Utilities Update #6
Reference Manual
part number 30000-90044
December, 1980

This update contains many error corrections and changes for MPE IV, including a clarified procedure for running SADUTIL and documentation for a new utility, DPAN4, which replaces DPAN2.

RJE/3000 Remote Job Entry Update #1
Reference Manual
part number 30000-90047
February, 1981

This update reflects a change in RJE/3000's mode of operation: The subsystem need no longer be run in Privileged Mode. Also, documentation of the CS Trace facility and the CS error messages has been brought up to date.

MRJE/3000 Multileaving Remote Job Entry Update #1
Reference Manual
part number 32192-90001
February, 1981

A new command, MRJECONTROL CHECK, is documented in this update. The command will permit a user who has been authorized to use the MRJECONTROL command to check the fix level of each MRJE module. Output from the host will now be owned by the user.account of the submitter. This means that listings will now have headers that identify the submitter.

COBOL/II Reference Manual Update #1
part number 32233-90001
July, 1980

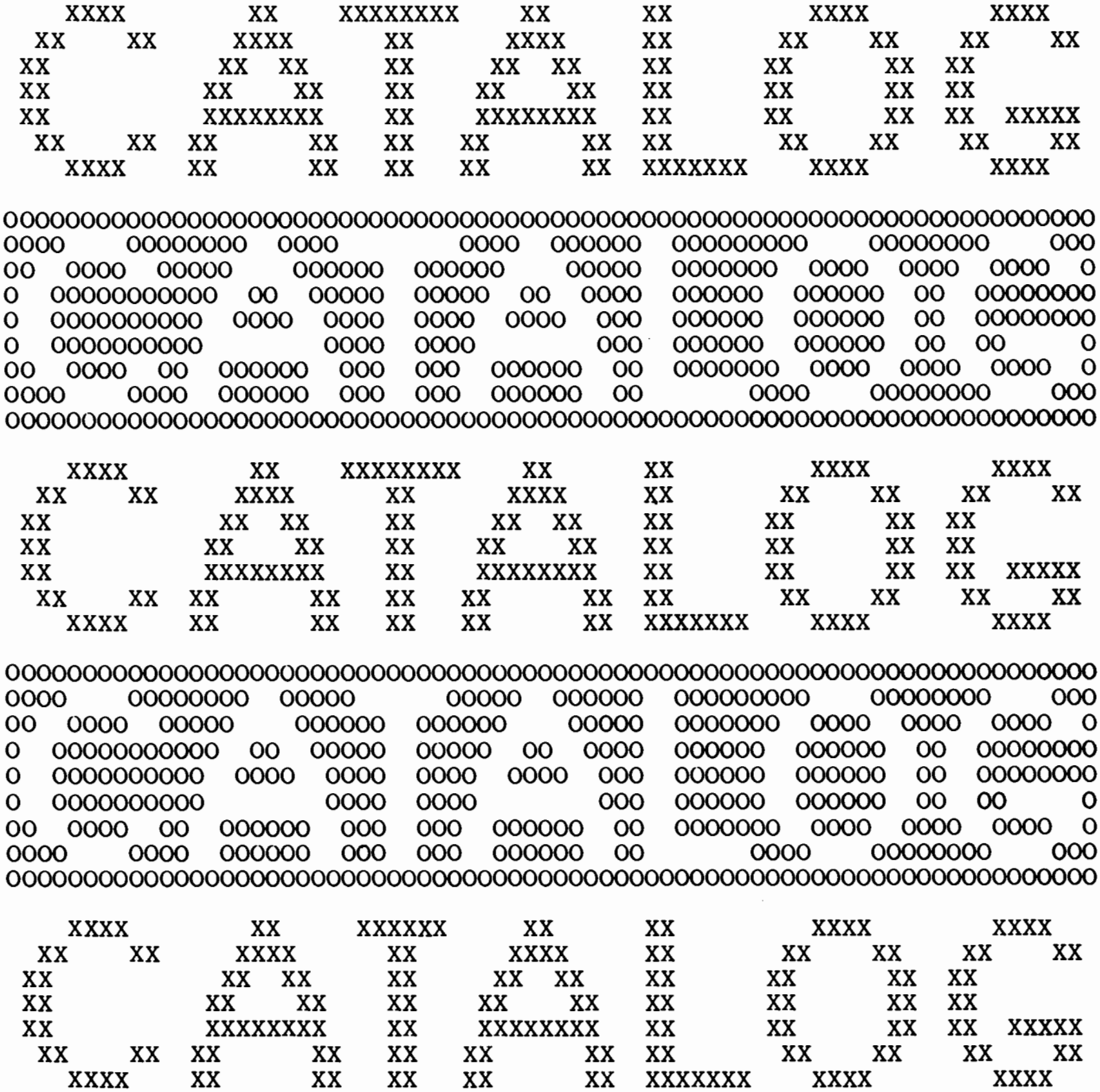
This update consists largely of clarifications of and corrections to the first edition. In addition, Appendix A documents two enhancements to the $CONTROL preprocessor command: a new ANSISUB option and a greatly expanded STDWARN option. Appendix C reflects a change to the compiler listing, and Appendix D provides information on determining the severity of compiler error messages.

Standard Product Costing
part number 32260-90009
February, 1981

In addition to several changes and corrections, this update documents the Financial Interface enhancement to Materials Management/3000.

System Customization Update #1
part number 32260-90011
February, 1981

This update includes expanded information about Materials Management/3000 batch jobs. All known problems have been corrected.



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 III 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	8.50	4/79	
	General Information Manual (Series II/III)		30000-90008	5.25	9/79	
*	MPE Commands Reference Manual		30000-90009	16.75	1/81	
*	MPE Intrinsics Reference Manual		30000-90010	20.00	1/81	
	MPE Segmenter Reference Manual	#	30000-90011	3.50	2/77	
*	MPE Debug/Stack Dump Reference Manual	#	30000-90012	4.50	9/76	10/8
*	Series II/III Console Operator's Guide		30000-90013	7.50	3/80	12/8

SYSTEM MANUALS (continued)

Manual Title	V	Part Number	Price	Print Date	Up-dated
*System Manager/System Supervisor Manual		30000-90014	11.75	2/81	
Error Messages and Recovery Manual		30000-90015	18.50	###	
HP 3000 Computer System Machine Instruction Set		30000-90022	6.75	2/80	
*MPE III System Utilities Reference Manual		30000-90044	4.50	3/77	1/81
Index to MPE Reference Documents		30000-90045	4.00	###	
*Software Pocket Guide		30000-90049	7.75	1/81	
Using Files	#	30000-90102	8.50	4/78	
*Series 30/33 Console Operator's Guide		30070-90025	12.75	12/80	
*Series 44 Console Operator's Guide		30090-90013	@	1/81	

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	4.25	6/76	2/77
Compiler Library Reference Manual		30000-90028	8.50	11/76	
FLEXIBLE DISCCOPY/3000		32199-90001	6.00	8/80	
SORT Reference Manual	#	32214-90001	5.50	3/80	

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	7.50	6/76	1/77
BASIC Interpreter Manual		30000-90026	13.00	6/76	8/78
FORTRAN Reference Manual		30000-90040	10.00	6/76	5/79
SPL Pocket Guide	#	32100-90001	2.00	11/76	
FORTRAN Pocket Guide	#	32102-90002	2.50	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
*Extra Function Sort for RPG/3000	#	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	3.25	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	

HP 3000 COMPUTER SYSTEMS

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	2/81
*Communications Handbook		30000-90105	13.50	2/81	
HP 30010A Intelligent Network Processor (INP) Installation & Service Manual		30010-90001	4.75	10/79	6/80
*HP 30010A/30020A Intelligent Network Processor Diagnostic Procedures Manual		30010-90002	4.25	2/81	
*HP 30020A Intelligent Network Processor (INP) Installation & Service Manual		30020-90001	4.50	10/79	
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
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	2/81	
DS/3000 to DS/1000 Reference Manual for HP 3000 Users		32190-90005	7.25	1/78	
*MRJE/3000 Reference Mnl.		32192-90001	10.50	8/80	2/81
*MTS/3000 Reference Mnl.		32193-90002	8.00	11/80	
*IML/3000 Reference Mnl.		32229-90001	12.00	11/80	

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	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	
*HP V/3000 Ref. Manual		32209-90001	14.50	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	11.75	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	55.00	7/80	

ADDITIONAL MANUALS (continued)

Manual Title	V	Part Number	Price	Print Date	Up-dated
Series 30 Installation Manual		30080-90001	6.25	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	2.70	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