

MOE

RECEIVED

JAN 3 - 1978

Volume 1 No. 4 November/December 1977

SHARAD HPDA

# Journal

of the HP 3000 Users Group

## Executive Board

### Chairman

William Bryden  
San Bernardino Valley Municipal  
Water District  
1350 South "E" Street  
San Bernardino, CA 92408  
(714) 889-0433

### Records

Barbara Rahe  
Teledyne Systems Company  
19601 Nordhoff Street  
Northridge, CA 91324  
(213) 886-2211

### HP Interface

Tom Harbron  
Anderson College  
Computing Center  
Anderson, IN 46011  
(317) 644-0951

### Computer Usage

Gil Drynan  
P. O. Box 313  
Woodinville, WA 98072  
(206) 773-8114

### Library

Gary D. Anderson  
Dept. of Epidemiology & Biostatistics  
McMaster University  
Hamilton, Ontario  
Canada L8S 4J9  
(416) 525-9140, Ext. 2434

### Meetings & Regional Users Group

Gil Drynan  
P. O. Box 313  
Woodinville, WA 98072  
(206) 773-8114

### Publications and Journal

Gary Green  
Research Coordinating Unit  
Maryland Dept. of Education  
P. O. Box 8717  
Baltimore, MD 21240  
(301) 796-8300

### 1978 Meeting Host

Joyce Pleasants  
Aurora Public Schools  
1085 Peoria Street  
Aurora, CO 80011  
(303) 344-8060

### Past Chairman

Bill Gates  
Longs Drug Stores, Inc.  
141 North Civic Drive  
Walnut Creek, CA 94596  
(415) 937-1170

### HP Representative

Ralph Manies  
Hewlett-Packard Company  
5303 Stevens Creek Blvd.  
Santa Clara, CA 95050  
(408) 249-7020

### Journal Editor

Elias Zabor  
HP 3000 Users Group  
c/o Hewlett-Packard Company  
5303 Stevens Creek Blvd.  
Santa Clara, CA 95050  
(408) 249-7020

## Inside this issue



Page

### Featured Articles

Structured Programming: State-Descriptive Systems, by Thomas R. Harbron . . . . .	2
Using Extra Data Segments, by Marc Hoff . . . . .	4

### Tips and Techniques

Get the Most Out of Your Printer, by Richard Starck . . . . .	5
Common Programming Errors With IMAGE/3000, by Robert M. Green . . . . .	6
Changing Compatibility Bits of a Program, by Robert M. Green . . . . .	6
COBOL-Formatted Data and Time From Other Languages, by Gregg Gloss . . . . .	7

### Contributed Library Corner

Martty Makela Wins HP67. . . . .	7
Contributed Library Contributions 20 & 21 . . . . .	7

### The Clearing House

IMACS . . . . .	8
COGO. . . . .	8
SPSSH. . . . .	8
Clearing House Response . . . . .	8

### All About Us

1978 Meeting Planning Underway. . . . .	9
Library Committee Now Chaired by Wayne Holt. . . . .	9
Regional User Groups. . . . .	9
Meeting Reports . . . . .	10
Interface Committee's Spooler Questionnaire Results. . . . .	11
How About A New Year's Resolution! . . . . .	12
Journal Distribution and Closing Date Notice . . . . .	12
New Information-Line Number . . . . .	12
Record Attendance at HP 3000 Users '77. . . . .	12

### Copyright Protection

The information in this publication may not be photocopied or reproduced without the prior written consent of the HP 3000 Users Group.

Copyright 1977 by HP 3000 Users Group

Published Bi-Monthly

Contributions: Address the Journal Editor

This publication is for the express purpose of dissemination of information to members of the HP 3000 Users Group. The information contained herein is the free expression of members. The HP 3000 Users Group and Editorial Staff are not responsible for the accuracy of technical material. Contributions from Hewlett-Packard Co. personnel are welcome and are not considered to be construed as official policy or position by Hewlett-Packard Company.

## Featured articles

---

### Structured Programming State — Descriptive Systems

by Thomas R. Harbron  
Anderson College  
Computing Center  
Anderson, Indiana 46011

The Fundamental Principle of Structured Programming is that at all times and under all circumstances, the programmer must keep the program within his intellectual grasp. The standard and well known methods for achieving this have been well documented and can be briefly summarized as follows:

1. Top-down design and construction,
2. Limited control structures,
3. Limited scope of data structures.

As productive as these methods are they should not be viewed as the only ones that support the Fundamental Principle. In fact, these methods should be thought of as only the first steps down the road to better programs and programmer productivity. Doubtless future developments will cause our present programming methods to appear crude by comparison.

2

This presentation explores another method which supports the Fundamental Principle and is complementary to the methods enumerated above. This method is not as general as those above, but addresses a problem which other methods have not simplified: Data translation; specifically the problem of converting a character string to another representation or vice versa.

Traditionally, this problem has been dealt with by requiring fixed "formats" of various types which are handled by library routines in such languages as FORTRAN and COBOL. While these fixed formats are usually satisfactory in batch work, they become totally unworkable for "free format," interactive I/O.

As a simple example, consider the situation where a user is asked to enter a dollar amount, and the character string entered is to be converted to a 32-bit integer representing cents. The sole specification is that any reasonable representation of a dollar amount which can be represented by a 32-bit integer, is to be accepted and converted. As examples, the following are all acceptable:

Character String	Integer Value
32	3200
32.17	3217
\$1,234.56	123456
-\$12,345,678	-1234567800
+6	600
-.05	-.05

This is obviously a very "loose" specification, the kind that makes programmers very uncomfortable. After all, what is reasonable to one person may not be to another, and everyone knows that when the program is finished it will be rigid in separating acceptable data. Why not then begin with a rigorous specification written down for all to read?

The answer is that the specification would violate the Fundamental Principle of Structured Programming! That is to say the specification would be so complex as to escape the intellectual grasp of its readers, even if it should be correct. The skeptical reader is invited to attempt to write such a specification for this problem. The task is not impossible; the product, however, is likely to be of doubtful value.

Perhaps this is why we struggle so with specifications and that complete specifications are seldom written for all but the simplest tasks. The bald truth is that natural language is ill-suited for writing specifications. As a corollary to this theorem, the most concise and intellectually manageable form for the specification of an algorithm is the algorithm itself.



While no formal proofs exist for these assertions, and the vagaries of natural language probably preclude such, no counter examples have been discovered by the author.

Returning to the original problem, it is proposed to develop an algorithm using a "state-descriptive system." In such a system there exists a finite number of states; transitions from state to state are controlled by transition rules so that:

1. The state is uniquely determined at all times.
2. The rules for transition to the next set are unambiguous and complete at each state.
3. Progress from the starting state to the terminal state is guaranteed.

Such a system may be represented in several forms, most commonly as a matrix or by a flow diagram. For programming purposes the flow diagram is usually preferable. However, it should not be confused with the traditional flow-chart.

A flow diagram for this example is shown in Figure 1. The following conventions are observed:

1. Initialization is done at 
2. Final processing is done at 
3. The conditions for each transition are shown on the corresponding arrow with the following conventions:

**HP Computer Museum**  
**[www.hpmuseum.net](http://www.hpmuseum.net)**

**For research and education purposes only.**

- a. Transitions are predicated on the value of the next character except for errors.
  - b. The letter N indicates the transition is to be made if the next character is numeric.
  - c. The letter T indicates the transition is to be made if the next character is the terminal character for the string.
  - d. Symbols enclosed by quote marks indicate the transition is to be made if the next character equals the quoted symbol.
  - e. No symbol indicates that the transition is unconditional.
  - f. When the conditions for a transition are not met, an abort occurs.
4. At each transition, the system proceeds to the next character. Thus the system always proceeds to termination.
  5. The processing at each node must be specified.

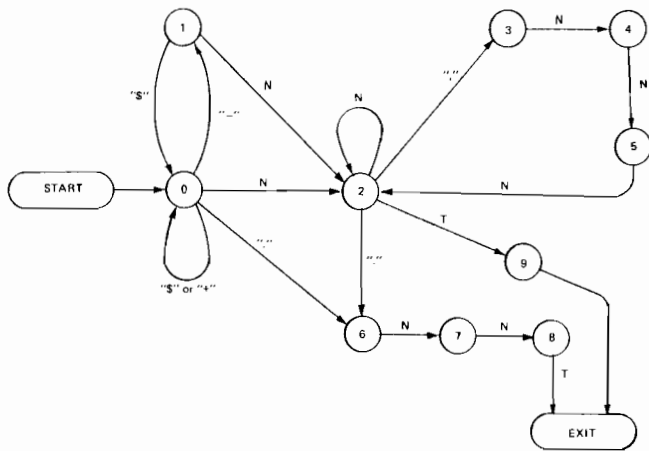


Figure 1

The following is an SPL (ALGOL) listing of a procedure for the example in Figure 1.

It can be seen that the actual programming is a trivial matter after the state diagram is finished. As a bonus, this procedure sets a hardware indicator and returns a pointer as the value of AMT in case of an abort.

```

1  PROCEDURE DOLLARTM(BUF,AMT);
2  BYTE ARRAY BUF;
3  DOUBLE AMT;
4  BEGIN
5  INTEGER P:=1,
6  STATE:=0;
7  BYTE TERM:=#15; <<CARRIAGE RETURN>>
8  LOGICAL EXIT:=FALSE,
9  ABORT:=FALSE,
10 NEG:=FALSE,
11 STAT:=Q-1;
12 SUBROUTINE BITM(B,A,ABORT);
13 BYTE B;
14 DOUBLE A;
15 LOGICAL ABORT;
16 IF A>2147483630 THEN <<O FLOW>> ABORT:=TRUE
17 ELSE A:=A+100*DOUBLE(B-#8);
18 <<END OF SBR T BITM>>
19
20 AMT:=0D;
21 DO BEGIN
22 CASE STATE OF BEGIN
23 <<0>>
24 IF BUF(P+1)=NUMERIC THEN STATE:=2
25 ELSE IF BUF(P+1)="S" OR BUF(P+1)="+" THEN STATE:=0
26 ELSE IF BUF(P+1)="N" THEN STATE:=1
27 ELSE IF BUF(P+1)="." THEN STATE:=6
28 ELSE ABORT:=TRUE;
29 <<1>>
30 BEGIN
31 NEG:=NOT NEG;
32 IF BUF(P+1)=NUMERIC THEN STATE:=2
33 ELSE IF BUF(P+1)="S" THEN STATE:=0
34 ELSE ABORT:=TRUE;
35 END;
36 <<2>>
37 BEGIN
38 BITM(BUF(P),AMT,ABORT);
39 IF BUF(P+1)=NUMERIC THEN STATE:=2
40 ELSE IF BUF(P+1)="." THEN STATE:=3
41 ELSE IF BUF(P+1)="+" THEN STATE:=6
42 ELSE IF BUF(P+1)=TERM THEN STATE:=9
43 ELSE ABORT:=TRUE;
44 END;
45 <<3>>
46 IF BUF(P+1)=NUMERIC THEN STATE:=4
47 ELSE ABORT:=TRUE;
48 <<4>>
49 BEGIN
50 BITM(BUF(P),AMT,ABORT);
51 IF BUF(P+1)=NUMERIC THEN STATE:=5
52 ELSE ABORT:=TRUE;
53 END;
54 <<5>>
55 BEGIN
56 BITM(BUF(P),AMT,ABORT);
57 IF BUF(P+1)=NUMERIC THEN STATE:=2
58 ELSE ABORT:=TRUE;
59 END;
60 <<6>>
61 IF BUF(P+1)=NUMERIC THEN STATE:=7
62 ELSE ABORT:=TRUE;
63 <<7>>
64 BEGIN
65 BITM(BUF(P),AMT,ABORT);
66 IF BUF(P+1)=NUMERIC THEN STATE:=8
67 ELSE ABORT:=TRUE;
68 END;
69 <<8>>
70 BEGIN
71 BITM(BUF(P),AMT,ABORT);
72 IF BUF(P+1)=TERM AND NOT ABORT THEN EXIT:=TRUE
73 ELSE ABORT:=TRUE;
74 END;
75 <<9>>
76 IF AMT>2147483630 THEN <<O FLOW>> ABORT:=TRUE
77 ELSE BEGIN
78 AMT:=AMT+100D;
79 EXIT:=TRUE;
80 END;
81 END; <<OF CASE STMT>>
82 P:=P+1;
83 END UNTIL EXIT OR ABORT;
84 IF NEG THEN AMT:=-AMT;
85 STAT,(612):=IF ABORT THEN 1 <<CCL >> ELSE 2 <<CCF>>;
86 IF ABORT THEN AMT:=DOUBLE(P);
87 END; <<PROCEDURE DOLLARTM>>
    
```

For the system shown in Figure 1, the processing rules are:

Node	Process
0	Null
1	Reverse sign flag
2	Convert digit by: 1. Multiply the object integer by 10 2. Add binary equivalent of digit to the object integer 3. Abort if overflow
3	Null
4	Convert digit
5	Convert digit
6	Null
7	Convert digit
8	Convert digit
9	Multiply object integer by 100, abort if overflow

The flowchart to implement this system is shown in Figure 2, and is similar to that for other state descriptive systems. At each iteration through the loop, the previously chosen state is selected, the processing appropriate to that state is done, the next state is selected, and the next character is moved into position for processing.

4

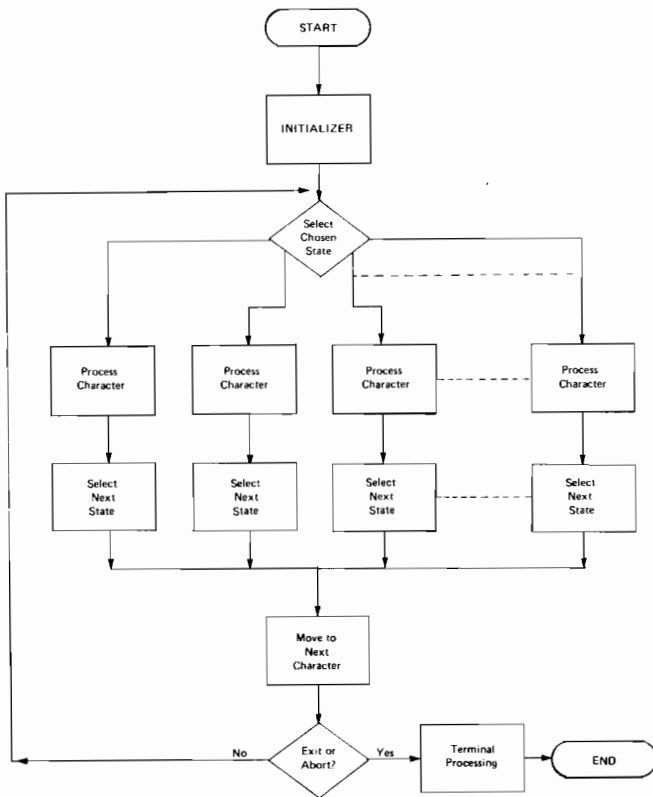


Figure 2

The advantage of this class of algorithms is in their manageability. The diagram in Figure 1 is easy to understand, when the conventions are known, and easy to modify. For example, the sequence of states 2, 3, 4, 5 permits commas as long as each comma is immediately followed by at least three digits. Thus 1,234 is allowed, but so is 1,2345. Should it be desirable to prohibit the latter case, the system could be modified by adding another state (10) following 5 as shown in Figure 3.

The processing rule for state 10 would be the same as for states 4 and 5. Since a state-descriptive system is intellectually manageable, it can be written in a minimum of time and will usually perform correctly the first time.

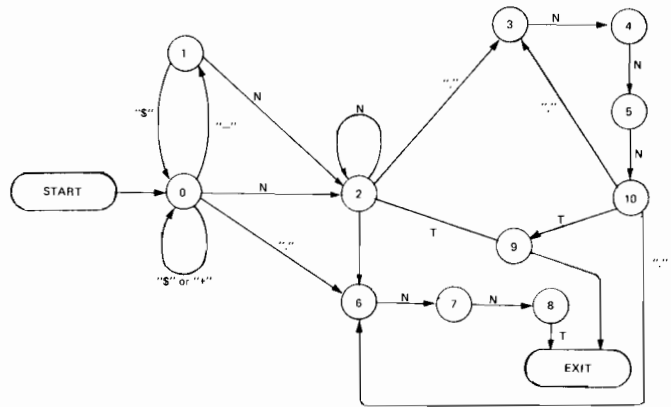


Figure 3

### Using Extra Data Segments

by Marc Hoff  
Hewlett-Packard Co.  
W 120 Century Rd.  
Paramus, New Jersey 07652

One of the special capabilities that may be assigned to an HP/3000 user is the ability to use extra data segments. This capability can be quite useful when one is writing system-level programs or when one wants to manage, in memory, more data than will fit in a user stack.

#### What is an extra data segment?

An extra data segment is a portion of memory, real or virtual that may be created by a user. It can be thought of as an extra stack, but has no registers associated with it, thus you cannot perform calculations directly on data stored in an extra data segment. Data must be moved between an extra data segment and your stack to be useful. One advantage over a stack is that an extra data segment can be either private or shared (stacks are always private). In addition, shared extra data segments can exist in a job or session after the program that created them has long terminated. A user program can have only one stack with a maximum of approximately 64K bytes but it can create and manage as many as 255 extra data segments (each up to 64K bytes).

**Extra Data Segments sound great — How do I use them?**

Many programmers use files to accomplish the same results that they could achieve with extra data segments. In fact, it is sometimes useful to think of extra data segments as temporary word-addressable random access files. There are five system intrinsics used to manage extra data segments, and they function very much like their file system cousins.

Data Segment Intrinsic	File System Intrinsic	Data Segment Function
GETDSEG	FOPEN (temp.)	Creates extra data segment
FREEDSEG	FCLOSE (purge)	Frees or expunges the segment
DMOVIN	FREADIR	Moves data to the stack
DMOVOUT	FWRITEDIR	Moves data to the data segment
ALTDSEG	*	Alters the logical size

\* Similar to using a logical end-of-file mark.

**Extra Data Segments can be used as I presently use files; Why should I switch?**

Extra data segments are much more efficient than files. If you currently use files, the logic is probably one-to-one replaceable, and the code generated is probably about the same length. Execution will be 25% to 900% faster and require proportionately less CPU time. Remember, the I/O system is transferring your data to and from the file in buffers that are also extra data segments.

**What about users who want to use a 500 x 500 array? How can they use extra data segments?**

Large array calculations rarely require more than one or two rows or columns of the array in memory at any given time. If the HP 3000 managed extremely large arrays for you, it could never be as efficient as you can be if you manage them yourself. In most cases, the logic to handle large array manipulations can be modified in a few minutes to use extra data segments. This time can then be easily recouped in the first few executions of the program.

**Using a shared extra data segment**

Most people can see the advantage of sharing extra data segments in a process-handling environment, but in a single-threaded environment they can also be useful. If you write a series of related but independent programs, you can use a shared extra data segment to communicate between them. In batch mode you can use that extra data segment to hold status words which would indicate if any previous programs aborted or completed successfully.

**Are there any pitfalls?**

Yes! Remember, programs that use extra data segments must be PREPped by a user with data segment capability, and ";CAP=DS" must be specified during the PREP. It is also important to remember that the number of extra data segments allowed per user is configurable during SYS-DUMP. When extra data segments are created they consume virtual memory. You may have to increase the amount of VM that you've configured in your system, and doing that requires a full reload.



**Tips and techniques**

**Get The Most Out of Your Printer**



by Richard Starck, President  
Ocean-Air International, Inc.  
RD #1 Burgettstown, PA 15021

Ocean-Air International has developed and is now marketing an interface controller that allows their two CPU's to share one line printer without operator intervention.

The controller automatically switches the printer between the two CPU's every forty seconds. Once a CPU requests and obtains control of the printer, it is maintained until forty seconds after the printing job is completed, at which time the printer is made available to the second CPU.

5

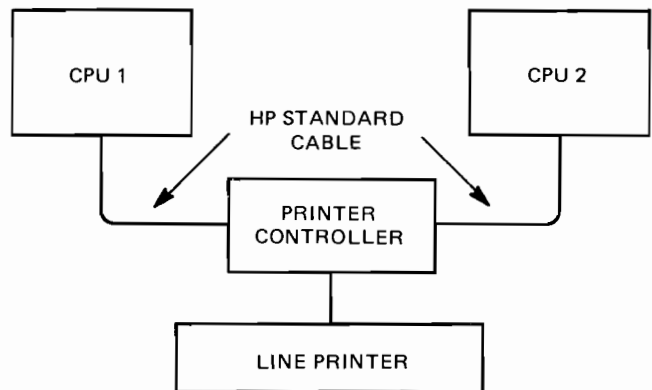
If a request is made by a CPU when the other CPU has the printer, a "device not available" message is sent to the console.

A three-position switch (CPU 1, AUTO, CPU 2) allows the operator to override the automatic feature and lock the printer to either CPU. This is necessary when using pre-printed forms.

LED indicators show which CPU currently has the printer.

Print spooling can be utilized on both CPU's.

The unit is constructed so that it can easily be bypassed in the event of failure or to eliminate any objections Hewlett-Packard might have in servicing either their line printer or line printer interface.



Sharing the printer in this manner has a number of benefits:

- 1) It eliminates the overhead imposed by DS/3000.
- 2) It keeps disc space utilization by spooled files to a minimum.
- 3) It is more economical than buying a second printer.
- 4) In an environment with multiple CPU's and line printers, multiple controllers can be used to balance printing loads thereby providing most efficient use of the line printers.

Several maintenance plans are available.

Printer controller and cable                      \$1,900.00  
 (Cable supplied connects controller to the line printer.) Both CPU's must have the standard line printer interface and cables in order to plug into the controller.

For further information, you may contact:  
 Richard Starck at (412) 681-7533.



### Common Programming Errors With IMAGE/3000

by Robert M. Green  
 President, Robelle Consulting, Ltd.  
 #130-5421 10th Avenue  
 Delta, B.C.  
 Canada V4M 3T9

6

Most of the programming errors I have seen people make with IMAGE/3000 fall into one of these two areas:

1. Problems with the "current" position.
2. Problems with locking.

#### Problems With the Current Position

The current position in an IMAGE dataset is important in most intrinsics: DBUPDATE and DBDELETE operate exclusively on the current position; DBGET (modes 2,3,5,6) read the "next" record relative to the current position. Many operations can change the "current" position and overlooking one of the cases is very easy.

1. You have done a DBGET on a manual master and are holding the data to update it later. You then do a DBFIND (with a different key value) on a detail dataset which is indexed by the same master dataset. This changes the current position in the master so that when you DBUPDATE the manual master you will get an ERROR 41 (critical item).
2. You are reading sequentially through a manual master (DBGET, mode 2). If you do *any* DBPUTs or DBDELETES as you are scanning, you may move your position and miss entries or read them twice. This is because records in manual masters are often moved as the result of a DBPUT or DBDELETE. The proper way to do this is to copy the key values into a file, then process from the file.

3. You are scanning a detail dataset (DBGET, mode 2). This works fine the first time, because the "current" position after DBOPEN is at the beginning of each dataset. However, the second time the program scans, it must rewind the dataset first (DBCLOSE, mode 2 or 3).
4. You have done a DBFIND on key value A. After a mode 5 DBGET, you DBDELETE the record, change the key to B and DBPUT it. How do you get the next record on the chain of A? You cannot just do a DBGET mode 5; you must first do another DBFIND with A. That is because the DBPUT changes the current position.

#### Problems With Locking

The basic problem with locking is forgetting to unlock the database; this can lock up all the terminals. There are several common ways to get into this kind of "deadlock":

1. You DBLOCK the database, but don't DBUNLOCK it in every case. The most frequent place to overlook is an error exit. This example is a good argument for structured procedures with only one entry and one exit. You lock upon entry and unlock upon exit.
2. Another source of lockups is the "BREAK" key. If the user hits this key while the program has the database locked, the database will remain locked until someone types :RESUME. The "BREAK" key can be disabled through FCONTROL and should be.
3. If the program opens two databases with locking, it must be :PREPPed with CAP=MR (multiple RINs). Unless the programmers are careful, it is easy to get into a deadlock where two terminals are eternally waiting for each other. This can be avoided by *always* locking the databases in a certain order and unlocking them in the reverse order.
4. All programs that lock a database should normally be run at the same priority level. This is especially serious on Pre-Series II machines where a low priority process can lock out high priority processes for an unlimited time.



### Changing Compatibility Bits of a Program (or, How to Stop Interactive Compiles)

by Robert M. Green  
 President, Robelle Consulting, Ltd.  
 #130-5421 10th Avenue  
 Delta, B.C.  
 Canada V4M 3T9

When a program is :PREPPed, the resulting program file is assigned certain capabilities:

:PREP \$OLDPASS, \$NEWPASS; CAP=PH,DS

The capability can only be assigned to the program file if the user and group have the capability (see LISTDIR program). In addition, the capabilities are rechecked at :RUN time; and those that can be assigned to a program are as follows:

PH	Process Handling
DS	Data Segments
MR	Multiple RINs
PM	Privileged Mode
IA	Interactive Access
BA	Batch Access

Each capability gives the program certain privileges that it would not otherwise enjoy. For example, MR allows a program to lock two data bases at the same time (and possibly deadlock with another program). IA allows the program to :RUN in SESSION-Mode (this is a default capability assigned to every program). The capabilities of a program can be checked with the LISTDIR program, and are stored in the first word of the first sector of a program file. Each capability is assigned a specific bit within the 16-bit word. The bit is turned on if the program has the capability. If the bits are numbered left to right as 0 through 15, then the following are the bits used:

PH	bit 15
DS	bit 14
MR	bit 12
PM	bit 9
IA	bit 8
BA	bit 7

It is possible to turn these bits on or off in order to dynamically give and take capabilities such as INTERACTIVE ACCESS. The following SPL/3000 program toggles (i.e., reverses) the IA bit in the COBOL compiler (COBOL.PUB.SYS):

```
BEGIN
  INTEGER FILENUM;
  BYTE ARRAY PROGRAMNAME (0:20):=
    "COBOL.PUB.SYS";
  ARRAY BUFFER (0:127);
  INTRINSIC FOPEN, FREADDIR, FWRITEDIR,
    PRINT'FILE'INFO;
  FILENUM:=FOPEN (PROGRAMNAME,1,4);
  IF < THEN PRINT'FILE'INFO (FILENUM)
  ELSE BEGIN
    FREADDIR (FILENUM, BUFFER, 128, 0 D);
    IF < > THEN PRINT'FILE'INFO (FILENUM)
    ELSE BEGIN
      BUFFER.(8:1):=NOT BUFFER.(8:1);
      FWRITEDIR (FILENUM,BUFFER,128,0 D);
      IF < > THEN PRINT'FILE'INFO (FILENUM);
    END;
  END;
END;
```

This program must be :RUN by MANAGER.SYS and will fail (on FOPEN) if the compiler is in use or :ALLOCATED.

The technique shown can also be used to give PH capability to EDIT/3000 (EDITOR.PUB.SYS) so that it can perform compiles (see Contributed Library) or switch priority sub-queues (GETPRIORITY intrinsic) by calling an external procedure.

## COBOL-Formatted Date and Time From Other Languages

by Greg Gloss  
Hewlett-Packard  
General Systems Division  
5303 Stevens Creek Blvd.  
Santa Clara, CA 95050

During the User's Group Meeting, a question was asked concerning the ability to call the COBOL Library routines for CURRENT-DATE and TIME-OF-DAY from other languages. If you have the COBOL run-time library in the SL, you can call the procedures from SPL and other languages. The CURRENT-DATE is returned in the form YY/MM/DD by the procedure C'DATE:

```
PROCEDURE C'DATE (YR'MO'DA);
  BYTE ARRAY YR'MO'DA;
  OPTION EXTERNAL;
```

The byte array YR'MO'DA must be at least 8 bytes long.

The TIME-OF-DAY is returned in the form HH:MM:SS by the procedure C'CLOCK:

```
PROCEDURE C'CLOCK (HH'MM'SS);
  BYTE ARRAY HH'MM'SS;
  OPTION EXTERNAL;
```

The byte array HH'MM'SS must be at least 8 bytes long.

• • •

## Contributed library corner

### Marty Makela Wins HP 67

The contributed library periodically awards a gift to a contributor. Marty Makela of Finland had the odds in his favor. Although several User Group Members submitted multiple contributions, Marty sent in six, which include the following:

- # 62 *Arithmetic Error Recovery Routines*
- #133 *Update to Stress System*
- #142 *COPYUSL: Copy RBMs from old USL to new USL-File*
- #149 *LIBRN3: Enhancement to Librarian Subsystem*
- #160 *LOADSL: Load Segments to SL from a USL-File*
- #164 *FINFILE: FTNOPEN*

Congratulations Marty!

Everyone who contributes has a chance to win. Send those neat routines in.

• • •

### Contributed Library Contributions 20 & 21

Any questions regarding this package should be directed to Don Gilchrist, McMaster University, (416) 525-9140, Ext. 2676.



## The clearing house

### IMACS — A Development and Maintenance Facility for Information Systems Using IMAGE

The Information Management and Control System, IMACS, is a self-contained information management system that dramatically reduces the time and effort needed to implement and maintain information systems that use the IMAGE data base management system. With IMACS it is possible to implement a data base and associated information system for data entry, update, retrieval and display in less than a day.

IMACS has three main components:

- An Information System Processor and Compiler that support a high-level, data base oriented language for the specification of information system programs.
- An extensive Data Dictionary and Directory System that can be accessed online by users, programmers, analysts and the data base administrator. The associated data base can also be accessed by application programs for data definitions, validation rules, etc.
- A Data Base Creation and Maintenance System that largely automates the creation, redefinition and re-structure of data bases.

The IMACS Information System Processor represents a totally new type of application system building facility to reach the commercial market. It is backed by over five years of research, development and application knowledge. The other two components contain facilities that automate many of the requirements of data base administration and therefore represent tools by which a data base administrator can document and control the data base resource.

The following prices represent a one-time license fee:

IMACS Information System Processor and Compiler . . . . .	\$ 15,000
— Data Dictionary and Directory System option . . . . .	\$ 3,000
— Data Base Creation and Maintenance System option . . . . .	\$ 2,000
Stand-alone Data Dictionary and Directory System . . . . .	\$ 8,000
— Data Base Creation and Maintenance System option . . . . .	\$ 2,000
Stand-alone Data Base Maintenance System . . . . .	\$ 5,000

Discounts are available for multiple installations within a single corporation. The fees include a two-year enhancement and maintenance contract after which time the contract becomes optional on an annual fee basis. Installation and education seminar options are available at additional charge.

IMACS is scheduled to be released for the HP 3000 computer system during December, 1977.

For an explanatory brochure and further details, please contact: D. C. Dummer & Associates, Ltd.  
40 Lake Lucerne Close S.E.  
Calgary, Alberta, Canada T2J 3H8  
(403) 261-4777



### COGO — (Coordinate Geometry for Engineers)

This is a program converted for use with the HP 3000 System, and is available from the original author:

James R. Requa  
Terra Graphic  
12 Bond Street  
White Plains, N.Y.  
Tel. (914) 948-0338 Price: \$500.00



### SPSSH

SPSSH is an authorized version of SPSS, the well-known package of statistical programs commonly used for research and instruction in the social sciences. SPSSH is designed to provide a meaningful statistical analysis capability to users of Hewlett-Packard computers. While SPSS and SPSSH were designed to serve the research and instructional needs of social scientists, both have proven useful to a large number of academic disciplines and professions.

SPSSH is designed so that the experienced SPSS user will immediately recognize virtually all the commands, and the user with even very limited computer experience will rapidly learn to perform various statistical operations. However, unlike SPSS, which is batch-oriented, SPSSH is designed to take advantage of the time-sharing capabilities of Hewlett-Packard computers.

SPSSH was developed at DePaul University for the HP 2000. It has been modified for the HP 3000 at the University of Wisconsin-River Falls. Both sites are committed to continual improvement and enhancement of SPSSH.

For additional information on SPSSH, contact one of the following:

Nicholas Elliott Political Science Dept. Univ. of Wisc.-River Falls River Falls, WI 54022 (715) 425-3318	David Feinstein Assoc. Director for Academic Computing Univ. of Wisc.-River Falls River Falls, WI 54022 (715) 425-3582
Nestor Dyhdalo Academic Computer Serv. DePaul University 2323 N. Seminary Ave. Chicago, IL 60614 (312) 321-8336	



### Correction to Clearing House Response Vol. 1, No. 1 May/June 1977

The following entry is corrected to update the personal contact only.

Manufacturing Application  
Package: Manufacturing Package for HP 3000  
Price: Varies  
Bob Wagster, Boeing Computer Services  
P. O. Box 24346, Seattle, WA 98124  
Tel. (206) 773-2267

## All about us

### 1978 Meeting Planning Underway

The 1978 International Meeting will be held from October 30th to November 3rd, 1978 in the Denver Hilton (located in downtown Denver, Colorado). Denver provides a convenient location; the Hilton offers excellent meeting facilities—and the view is breathtaking: the Rocky Mountains to the West and the city to the East.

As our meetings get larger, facilities, early planning and lots of member participation will be essential for an educational, professional and enjoyable meeting. More than 300 attendees participated in the Seattle Meeting; with the contingency growth of the HP 3000 installed base, the Denver meeting attendance will more than double the Seattle count.

Under the guiding hand of Joyce Pleasants, the 1978 Meeting Host, excellent progress is being made. In addition, at a recent Rocky Mountain Regional Users Group meeting, committees were formed and staffed for Program, Registration, Exhibits and Activities.

At the forthcoming Director's Meeting in January, preliminary plans will undergo review and approval; from that point, hundreds of individual activities that make a successful meeting, will begin—and these will culminate in the Group's largest, most successful meeting for you.

*Mark your calendar now* (while the information is still fresh in your mind!); let Joyce Pleasants know your ideas—(303) 344-8060, let her know how you can participate in the 1978 meeting.



### Library Committee Now Chaired by Wayne Holt

Wayne Holt, Chairman of the Library Committee has had some good inputs from committee members and other participants as to methods to make the library more useful. In the past year, the goal of more frequent distributions was achieved.

The potential of the Contributed Library is tremendous; cost savings to users are quickly realized as soon as you use a program or routine. There is over \$100,000 (conservatively) of programming investment available for a mere \$100.00 annually, which is the cost of a site membership (and this includes one individual membership).

Wayne is anxious to get your ideas on how the library can:

- obtain more contributions
- be more useful to newer users
- be distributed on a timely and economical basis

If you have any ideas, contact Wayne Holt at Box F69, Whitman College, Walla Walla, WA 99362. Telephone (509) 527-5417.

## Regional User Groups Listing

The following list represents the total information available to your editor to date. If any data is incorrect, or a RUG is unlisted, please advise and the information will be updated with the next issue of the Journal.

### Regional User Groups (North America)

Regional Group	Contact
Baltiwash Baltimore/Washington	Dick Neumyer, Dept. 8850 Western Electric Co. 2500 Broening Highway Baltimore, MD 21124 Tel. (301) 675-9000  1st Wednesday of every odd-numbered month Next Meeting January 4, 1978 At HP, 4 Choke Cherry Road Rockville, MD 20850
RMRUG Rocky Mountain	Charles R. Van Ausdall Commercial Office Products 4905 Lima Street Denver, CO 80239 Tel. (303) 373-4320
BARUG Bay Area/San Francisco	Bill Gates Longs Drug Stores, Inc. 141 North Civic Drive Walnut Creek, CA 94596
Calgary/Western Canada	Alex Carrughers Hewlett Packard Co. Suite 210, 7220 Fisher St. S.E. Calgary Alberta Canada T2H 2H8  Now forming - please call/write, join
CCRUG Central Canada (Toronto)	Doug Wilson Conestoga College 299 Doon Valley Drive Kitchener, Ontario Canada N2G 3W5 (Tel.) (519) 653-2511  2nd Tuesday of every odd-numbered month Next Meeting January 10, 1978 At HP, 6788 Goreway Drive Mississauga Ontario Canada L4V 1M8
HOURUG Houston, Texas	Glen Godwin Paktank Corp. - Suite 1800 2000 West Loop South Houston, TX 77027
NEORUG (Northeast Ohio) Cleveland, Ohio	Mike Vislosky, Room 300 Mandate Corporation 1717 East 9th Street Cleveland, OH 44114

- NOWRUG  
Northwest (Seattle, WA)      Lawrence F. Mahoney  
   R. W. Beck & Associates  
   200 Tower Building  
   Seattle, WA 98101
- GNYRUG                              Bill Clansman  
Greater New York                AIRCO Industrial Gasses  
   P. O. Box 1601  
   Union, NJ 07083
- SCRUG                                Bill Bryden  
Southern California            San Bernardino Valley  
   Water District  
   P. O. Box 5906  
   San Bernardino, CA 92412  
   (714) 889-0433

Next Meeting March 1st and 2nd  
At Queen Mary Long Beach, CA

### REGIONAL USER GROUPS (EUROPE)

- Belgium and Holland      Louis Soudan  
   Antwerpse Bounerken Verbeeck, N.V.  
   Bouwensstraat 29-35  
   2200 Borgerhout  
   Belgium
- Germany                            Jochen Gomann  
   Schmalbach Lubeca  
   Schmalbachstrasse 1  
   D-3300 Braunschweig  
   Germany
- Switzerland                    Dr. Emil R. Brandli  
   Technikum Winterthur  
   Ingenieurschule  
   CH 8401 Winterthur  
   Schweiz (Switzerland)

March 3, 1978  
Vevey (Switzerland)

- Scandanvia  
(Sweden, Finland,  
Denmark)                    Esko J. Vuohu  
   Geological Survey of Finland  
   DP - Group  
   02150 Espoo 15  
   Finland
- U.K.                                 Mr. C. A. Satterthwaite  
   FM Insurance Company, Ltd.  
   Kingsgate House  
   66-74 Victoria Street  
   London SW1E 6SQ  
   U.K.
- France                              Mr. J. P. Aeberhardt  
   Credit Lyonnais  
   Groupe De Recherches  
   Operationnelles  
   10 rue Saint Augustin  
   75002 Paris  
   France
- Italy                                 Dino Ferretti  
   20131 Milano Via Stradivari 4  
   Italy

### Meeting Reports

Under this subheading, you will find henceforth capsule reviews of meetings that have been held. Please bear in mind, the deadline for receiving this information is 4 weeks prior to publication. In your review writeup, tell us the RUG name, meeting date, location and time, topics covered, your name, phone, etc.

### Baltiwash RUG Met November 2, 1977

by Bob Chaffin  
HP Eastern Technical Center  
4 Choke Cherry Road  
Rockville, MD 20850

Attendees at the Rockville District Sales office heard about the Seattle meeting from Craig Jester - HP Rockville SE, John Borden - Baltimore Gas & Electric HP 3000 System Manager, and Bob Chaffin - HP Eastern Technical Center SE. The meeting was hosted by Jack Roof of the Maryland National Park and Planning Commission. RUG president Dick Neumyer of Western Electric introduced the first speaker, Craig Jester, who discussed the MPE IIB software enhancements that had been previously discussed at the International Users Meeting in Seattle. John Borden then presented a summary of discussions he observed at the KSAM, IMAGE, RJE, APL and SPOOLING seminars in Seattle. Bob Chaffin reviewed with the group a large number of his notes, taken at sessions he attended at the Seattle meeting. The meeting closed at 10:00 p.m. with more than 40 people signed in. Next meeting will be January 4, 1978.

### Interface Committee's Spooler Questionnaire Results

The results have been compiled for the Interface Committee questionnaire, distributed to members attending the International Meeting at Seattle.

Each item was rated and prioritized according to the following reference numbers:

- |                   |  |
|-------------------|--|
| <b>Rating</b>     | <b>Priority</b>                            |
| 1 – Necessary     | 1 – Necessary . . . . 10 – Not Important   |
| 2 – Important     | (on a sliding scale in descending order of |
| 3 – Nice          | importance from 1 to 10)                   |
| 4 – Not Important |  |

The raw data follows; the figures are in percent of responses to the question (rounded to the nearest %). Response accounted for 15.7% of the users attending the Seattle Meeting (47 returned responses).

#### Spooler Enhancements Questionnaire – RESULTS

Rating				Priority			Enhancement
1	2	3	4	1-3	4-6	7-10	
64%	21	6	9	77	12	11	1. Ability to back-up printing by a) a number of pages b) a number of lines
27	12	33	28	40	24	36	
							2. Number of copies (check one) a) 64 copies are adequate 91.2% b) an upper limit is desired 8.8%
15	20	29	36	27	32	41	
10	26	40	24	32	32	36	3. Allow already printed active spool file data (extents) to be purged or not purged as a) an installation option b) a job or file parameter option c) a spool command for each device under spool control
14	21	36	29	27	43	30	
37	27	35	1	53	38	9	4. Forms type indicator in "showout" command
47	21	23	9	64	20	16	5. Should similar "forms" be printed consecutively (together)?
11	13	43	33	28	30	42	6. Header trailer enhancements a) block characters b) "*" on perforation to show split between jobs c) accounting statistics on spool output – lines printed – CPU seconds – elapsed time
6	17	47	30	16	33	51	
17	39	30	14	42	40	18	
9	38	34	19	33	37	30	
9	34	33	19	30	41	29	
10	32	47	11	31	40	29	7. Prioritize output by the number of printlines so that shorter jobs have better turnaround.
8	31	40	21	34	25	41	8. Be able to repeat output on some other device
23	27	38	12	45	45	10	9. SHOWOUT to show number of printlines already printed on active output
17	21	38	24	32	29	39	10. Spook a) ability to run as batch job b) edit a spoolfile by deleting a range of lines, records or pages c) convert spoolfile to permanent fixed length file, accessible by file system utilities. Give it standard file name
22	50	17	11	56	30	14	
30	45	16	9	54	31	15	

Additionally, many comments were received; the total information package was submitted to the interested parties at HP General Systems Division.

### How About A New Year's Resolution!

A new year is almost upon us, and we want you to reward yourself. How? Resolve to (pick one or more):

- Submit an article, a tip or technique to the Journal
- Give a talk or lead a session at a Regional Users Group Meeting
- Actively participate in the 1978 meeting by giving a presentation, helping out in a panel discussion
- Submit that neat routine of yours to the Contributed Library

We promise it will be rewarding to you – because if just one person other than yourself does that, you've doubled your money.



### Journal Distribution and Closing Date Notice

The intent is to henceforth distribute the Journal during the first week of the second calendar month of each issue. Consequently, beginning with the January/February 1978 issue, distribution will be made the first week in February; the closing date for contributions is four weeks prior. Please send your contributions and help us observe these deadlines.

12

Issue	Distribution	Contributions Closing Date
Jan/Feb	1st week in Feb	1st week in Jan
Mar/April	1st week in April	1st week in Mar
May/June	1st week in June	1st week in May
July/August	1st week in August	1st week in July
Sept/October	1st week in Oct	1st week in Sept
Nov/Dec	1st week in Dec	1st week in Nov

#### CHANGING YOUR ADDRESS?

##### Please Let Us Know!

Cut the label portion of the envelope in which you receive the Journal (or duplicate that information on a sheet of paper) and mail it *with your new address* to the Journal Editor (address on cover).

This is especially important if a site membership replaces the designated voting member. That's the individual to whom library tapes are addressed.

### New Information-Line Number: (408) 255-1010

The Information-Line is your 24-hour number to the HP 3000 Users Group Central Desk.

To pass-on tips or request information – call the number. When you call, the system will deliver a 30 to 60-second message. Then, speak as long as you want – the system is voice actuated. Be sure to give your name, phone number and location.



### Record Attendance at HP 3000 Users '77 Seattle Conference

Over 320 users, representing more than 160 companies, attended the HP 3000 Users Group 1977 meeting. The theme of the conference was "Interface and Education." Attendees had a chance to get into in-depth discussions with other users, 20 GSD representatives, and the 7 field systems engineers who attended. Topics ranged from installation management (or "How I run my shop" experiences) to advanced programming techniques on languages and data management. There were more than 40 technical sessions and all were well attended. Many additional informal discussions lasted, in many cases, until the early morning hours. An on-site system (compliments of the Neely Bellevue office) was employed for user software demonstrations. The excitement and intensity of attendees can be best measured by the use of the system – if you stopped by the system at midnight, you would have seen people still sharing ideas and experiences.

From all reports, it was the best users group meeting yet. Many people, both users and HP, jointly contributed to the success of the meeting. And both users and HP gained new ideas and insights. As one GSD lab representative put it, "The chance to talk with users helps me get a much better understanding of how the system is used, and gives me a background that means I'll be able to design a better product." Similarly, users gained insight – as one user put it, "I've learned a lot from talking with HP people and other users. The investment of this week will save my shop at least a couple of man-years of effort. For sure I'm one that will be attending the Denver meeting."

The HP 3000 Users Group meeting for 1978 is already scheduled. Place: Denver, Colorado; dates: October 30 to November 3, 1978. Plans are already under way for another exciting and outstanding conference.