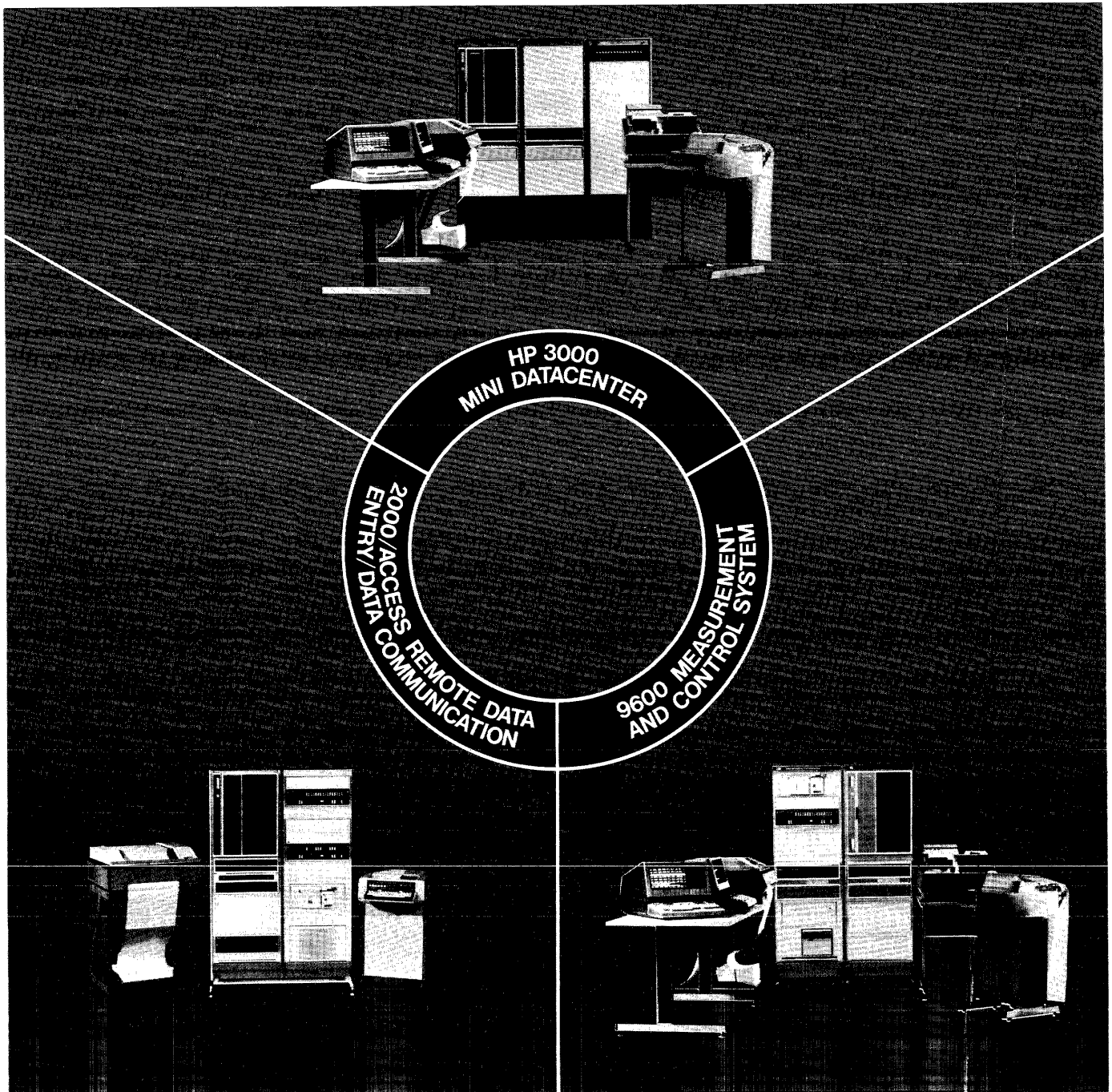


ISSUE No. 8
MAY 1976

computer systems COMMUNICATOR



subscription information

Annual subscriptions consisting of 6 issues are available as outlined below.

I. CUSTOMERS WITH SOFTWARE MAINTENANCE AGREEMENTS OR SOFTWARE SUBSCRIPTION SERVICE AGREEMENTS (SOFTWARE SERVICE CONTRACT SUBSCRIPTIONS)

All Hewlett-Packard customers with Software Service Contracts are entitled to one BASE SUBSCRIPTION (1 copy per issue) at no additional charge. These customers may also buy ADDITIONAL SUBSCRIPTIONS whose purchase price is to be prorated against the remaining life of their Software Service Contract. A proration table appears on the ORDER FORM which is bound into this issue.

To receive a BASE SUBSCRIPTION at no charge as well as to purchase ADDITIONAL SUBSCRIPTIONS under the provisions of the Software Service Contract Program, complete the ORDER FORM and forward it to your local HP Sales and Service Office. Your local Customer Engineer will validate your order and mail it to the appropriate HP department.

Rates:	U.S.A.	NON-U.S.A.
BASE SUBSCRIPTION	NAC*	NAC*
ADDITIONAL SUBSCRIPTIONS (ea.)	\$12/yr.	**

- 1) ADDITIONAL SUBSCRIPTIONS must go to the same name and address as the BASE SUBSCRIPTION to qualify for the reduced rates.
- 2) ADDITIONAL SUBSCRIPTIONS ordered at a later date than the BASE SUBSCRIPTION must include, with the order form, a copy of the address label for proper identification.
- 3) Charges for ADDITIONAL SUBSCRIPTIONS will be prorated to expire with your Software Service Contract.
- 4) Orders for ADDITIONAL SUBSCRIPTIONS from a customer with a Software Service Contract will be verified by the Customer Engineer who will complete the "FOR HP USE ONLY" portion of the subscription form and direct the order to the appropriate HP department. The customer will be billed by his local HP Customer Engineering Department.

*No Additional Charge (NAC)

**Contact your local HP Customer Engineer for the price in the currency of your country.

II. CUSTOMERS WITHOUT SOFTWARE MAINTENANCE AGREEMENTS OR SOFTWARE SUBSCRIPTION SERVICE AGREEMENTS (MAIL ORDER SUBSCRIPTIONS)

Rates:	U.S.A.	NON-U.S.A.
BASE SUBSCRIPTION	\$48/yr.	***
ADDITIONAL SUBSCRIPTIONS (ea.)	\$12/yr.	***

- 1) ADDITIONAL SUBSCRIPTIONS must be ordered at the same time as the BASE SUBSCRIPTION and go to the same name and address as the BASE SUBSCRIPTION to qualify for the reduced rate.

- 3) Complete the ORDER FORM as directed and mail together with your payment to:

**Hewlett-Packard Co.
Mail Order Dept.
P.O. Drawer No. 20
Mountain View, California 94043
U.S.A..**

SUBSCRIPTION CORRESPONDENCE

Address all correspondence relating to **COMMUNICATOR** subscriptions to:

**Subscription Service Manager
Hewlett-Packard Company
Mail Order Dept.
P.O. Drawer No. 20
Mountain View, California 94043
U.S.A.**

***The international customer is encouraged to also use HP's Direct Mail Order System by remitting a bank draft in U.S. dollars according to the order procedure outlines above. If the currency regulations in the customer's country disallow the purchase of bank drafts in American dollars, or if the customer does not have ready access to the required banking services, the customer may order subscriptions from the local HP Sales and Service Office through his Customer Engineer. The customer should contact his HP Office for the price of the subscription in the currency of his country then complete the ORDER FORM and forward it together with payment to his local HP Office.

HEWLETT-PACKARD COMPUTER SYSTEMS COMMUNICATOR ORDER FORM

Please Print:

Name _____ Title _____
 Company _____
 Street _____
 City _____ State _____ Zip Code _____
 Country _____

☐ MAIL ORDER SUBSCRIPTIONS

BASE SUBSCRIPTION \$ _____

_____ ADDITIONAL SUBSCRIPTION(S) \$ _____
 No.

TOTAL AMOUNT ENCLOSED \$ _____

☐ SOFTWARE SERVICE CONTRACT SUBSCRIPTIONS

BASE SUBSCRIPTION (NO ADDITIONAL CHARGE) NAC _____

_____ ADDITIONAL SUBSCRIPTION(S) \$ _____
 No.

TOTAL AMOUNT YOU WILL BE BILLED \$ _____

FOR HP USE ONLY

SUPPORT OFFICE NUMBER _____ ORDER DATE _____
 APPROVED BY _____ C.E. NUMBER _____
 SERVICE CONTRACT NUMBER _____ EXPIRATION DATE _____
 AUTHORIZED TOTAL NUMBER OF SUBSCRIPTIONS _____
 CUSTOMER'S HP OPERATING SYSTEM _____

Printed 4/76

TABLE OF PRORATED \$ AMOUNT DUE PER ADDITIONAL SUBSCRIPTION

(Use only for ordering ADDITIONAL SUBSCRIPTION(S) against an existing Software Service Contract)

Months Remaining in Service Contract												
	1	2	3	4	5	6	7	8	9	10	11	12
Cost of Each ADDITIONAL SUBSCRIPTION	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	11.00	12.00

INSTRUCTIONS FOR ORDERING COMMUNICATOR

All Hewlett-Packard customers with Software Service Contracts are entitled to one BASE SUBSCRIPTION (1 copy per issue) at no additional charge. These customers may also buy ADDITIONAL SUBSCRIPTIONS whose purchase price is to be prorated against the remaining life of their Software Service Contract.

Customers who do not have Software Service Contracts may purchase Mail-Order Subscriptions through HP's Direct Mail Order System.

A. MAIL-ORDER SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
 2. Compute amount due:
 - a) Annual Base Subscription (6 issues) \$ 48.00
 - b) _____ Additional Subscriptions*
@ \$12.00 ea. \$ _____

 - c) Total Order Amount (a + b) \$ _____
 - d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Mail check or bank draft with ORDER FORM to:

HEWLETT-PACKARD CO.
MAIL ORDER DEPARTMENT
P.O. DRAWER #20
MOUNTAIN VIEW, CA. 94043
U.S.A.

B. SOFTWARE SERVICE CONTRACT SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
 2. Compute amount due: (BASE SUBSCRIPTION is at no additional charge.)
 - a) Annual Base Subscription (6 issues) \$ 0.00
 - b) _____ Additional Subscriptions*
\$ _____

Prorate the dollar amount to make the ADDITIONAL SUBSCRIPTIONS EXPIRE WITH YOUR Software Service Contract. (SEE TABLE)

 - c) Total Order Amount (a + b) \$ _____
 - d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Forward ORDER FORM to your local HP Customer Engineering Representative. Your order will be approved and forwarded to the appropriate department. You will be billed for any ADDITIONAL SUBSCRIPTIONS by your local HP office.

C. SPECIAL INSTRUCTIONS FOR INTERNATIONAL CUSTOMERS

- | | |
|--|---|
| <ol style="list-style-type: none">1. International customers who do not have Software Service Contracts are encouraged to use HP's Direct Mail Order System by remitting a bank draft in U.S. dollars according to the ordering procedures outlined in Instruction A above. Optionally, international customers may purchase the Communicator through their local HP Sales and Service Office. The customer should contact his HP Office for the subscription prices in the currency of his country, then complete the Order Form and forward it together with payment to his local HP Customer Engineering Department. | <ol style="list-style-type: none">1. International customers with Software Service Contracts should follow the ordering procedure outlined in Instruction B above. If the customer wishes to purchase ADDITIONAL SUBSCRIPTIONS, he should contact the local HP Office for the subscription price in the currency of his country, then submit the ORDER FORM. The customer will be billed for ADDITIONAL SUBSCRIPTIONS by his local HP Office. |
|--|---|
-

*All ADDITIONAL SUBSCRIPTIONS will be sent to the same name and address as the BASE SUBSCRIPTION.

editor's note



The publication date of this issue of the **Communicator** coincides with the introduction of the new HP 3000 Series II Computer Systems. You will find information describing the new system, and a complete summary of the new support services in the feature article of this issue.

As described in the feature article, software information service for the HP 3000 is provided in the form of a subscription to the **Communicator** published by General Systems Division six times per year. This change in the publication schedule (from eight issues per year to six issues per year) will not affect any of your subscriptions ordered before April 1, 1976. All subscribers who placed their orders before April 1, 1976 will receive eight issues of the **Communicator** as originally offered. All new subscriptions received *after* April 1, 1976 will cover a *year's* subscription to the **Communicator**, beginning with the next publication date after receipt of the subscription order. The changeover to the new bimonthly publication date takes place after the October 1, 1976 **Communicator**.

Your comments on our new support services, as well as other articles in the **Communicator** are always welcome.

Address correspondence to:

Editor, Computer Systems **Communicator**
Hewlett-Packard Company
General Systems Division
5303 Stevens Creek Blvd.
Santa Clara, Ca. 95050

contents

ABOUT THE HP 2000

Software Tips

HELLO! 342

Software Updates

HP 2000/Access, HP 22687A, Rev. 1615 342

Documentation 343

Training Schedule 345

ABOUT THE HP 3000

Software Tips

Tip on Using EDIT/3000 346

New Release of FORTRAN/3000 347

Bulletins

How to Obtain an Update to a Manual 349

New MPE/3000 Reference Manual Update 349

New FORTRAN/3000 Manual 349

Software Updates

BASIC/3000 350

BASICOMP/3000 350

COBOL/3000 351

FORTRAN/3000 351

RPG/3000 354

Documentation 355

Training Schedule 356

FEATURE ARTICLE

Introducing HP 3000 Series II Computer

Systems 359

A Complete Range of Support Services 360

ABOUT THE 9600/9700

Software Tips

Program Segmenting 365

RTE-II/III Class Table Structure 367

The WHZAT Program 368

"NO ABORT" Return from FORTRAN

Subroutines 370

Know Your RTE, Part 2 371

Software Sam 378

Software Updates

RTE-II/RTE-III 380

RTE Drivers 382

DOS-IIIB Modules 382

Documentation 383

Training Schedule 386

HELLO!

Gary Koerzendorfer
HP General Systems Division

"HELLO" is a special program on the HP 2000 Access System that resides in A000, the system manager account. HELLO is automatically executed when a user logs on to the system (regardless of the security level of HELLO). The typical use of HELLO is to give a greeting, print time, month, day, and year (possibly via the subroutine DATTIM listed in page 104 of **Communicator** Issue 3); and also to give information of interest to users.

Beyond these uses, HELLO has great potential from the system manager's point of view. For example, when the backup mag tape for the HP Data Center 2000 Access became excessive, a voluntary campaign to reduce user storage was instituted. A routine was added to HELLO that calculated and printed the amount of disc storage the account was using, accompanied by an appropriate message.

The ability to programmatically determine the ID-code of the user also brings to mind some valuable features. In a simple case, different messages may be output depending upon who is logging on. For example, the RJE schedule could be printed for only RJE users.

As a more advanced example, consider the following situation: an educational institution running Computer Aided Instruction (CAI). The school may want to limit the CAI students to running the instructional program during certain hours of the day, to lessen the terminal load, and also ensure that the CAI people run the CAI and not play spacewar on a CRT. The solution to this problem would be to assign specific ID-codes to the CAI students, and to have HELLO check for these ID-codes. If HELLO finds one of the ID codes, it would next check the time of day to see if the student was logging on during the allowed time. If the logon was at a disallowed time, HELLO could print a message to that effect and log the user off using the SYSTEM function with BYE as an argument. If the student was logging on at an allowed time, HELLO could directly invoke ("CHAIN" to) the CAI. If the break key was disabled during this process, the user would not be able to abort or circumvent this process. If the CAI program was so designed, the only way to stop it would be to enter an exit request, at which time the CAI could log off the user without any further action on their part (or despite it).

Also available to the system manager is a facility to do activity profile and accounting operations by including routines in HELLO. Such a routine could record the time, port number, and user ID for each logon.

A small routine may even be included in HELLO to calculate the time of day of the end of a session. When a

logon occurred, HELLO could take the total terminal time accumulated to that point, and by subtracting the total terminal time recorded at the last logon, the length of the last session can be found. If this is added to the time of day of the last logon, the logoff time will result. The last step in this procedure would be to record the current terminal time and time of day on disc to be used for this purpose at the next logon.

The requirements for logging, accounting, and system management vary as widely as the use of the systems themselves. For some installations, the built-in accounting features may be adequate, or even unnecessary. However, when they are inadequate, a routine in HELLO combining the security features of the 2000 Access with Access/BASIC can go a long way toward fulfilling these requirements.

Each issue of the **Communicator** will provide you with information pertinent to the status of 2000 software products, including the latest software changes and enhancements.

The software updates described in this issue relate to:

Product	Number	Revision Date Code
HP 2000/Access	HP 22687A	1615

Where changes in the documentation are indicated, updates to the appropriate manuals will be printed. This information is provided simply as a temporary measure.

HP 2000/ACCESS HP 22687A, Rev. 1615

The following problems were corrected in the **I/O Processor** in Revision 1615 of the 2000/Access System:

1. The IOP could go into an infinite loop or execute a HALT 2 if a terminal was mis-diagnosed as an IBM 2741.
2. A HALT 2 could have been executed because the interrupt system was not always on when expected.
3. There was a very high probability of executing a HALT 52 when the PHONES-0 command was given from the system console.

4. A HALT 75 could have been executed when signing off from an IBM host system.
5. IBM host control codes SKIP TO CHANNEL 9 through SKIP TO CHANNEL 12 for the line printer were incorrectly mapped into SKIP TO CHANNEL 3.

The following problems were corrected in the **System Processor** in Revision 1615:

1. The system would occasionally go into an infinite loop when an RJE connection was made.
2. The COPY command would make some amount of space unavailable to the system if a disc error occurred when reading the program or file.

The following are known problems in the system and will be corrected at some future date (known problems documented in the release of Revision 1614 also still exist):

1. Specification of a record length less than the maximum buffer length when using the FILE command with a JL or JP job function designator could cause the IOP to crash when that file is used.
2. Use of a multiplication operator in a CONVERT statement could cause a spurious error message. The statement 'CONVERT "1234" TO A(I*2)' would cause this problem.
3. The following program will cause a spurious error message:
(Assume the file named 123456 does not exist before running the program.)

```

10 DIM A$(7)
20 FILES *
30 A$="123456"
40 ASSIGN A$,1,R,"MASK",RR
60 CREATE R,A$,10
70 ASSIGN A$,1,R,"MASK",RR
80 END

```

The error occurs in line 70 and can be corrected by removing line 40.

There has been some trouble reported when using the 7260A card reader in tape mode with the Access system. You must have option G40 for the 7260A installed and use the single pick function on the 7260A to properly read a program in tape mode.

Enhancement to Access System in Revision 1615

Remember to reconfigure your IOP and to perform a system update followed by a SLEEP to magnetic tape to properly convert to Revision 1615. There is no patch tape for Revision 1615.

When communicating with an IBM host the RJE-SC and RJE-SD commands now allows tabbing to a column number when entering a SIGNON or SIGNOFF message. When entering the command you may type a control-T followed by the column position you wish the next character to be entered. For example, you would type 'RJE-SC,,/*SIGNON<cntrl-T>16REMOTE18<cntrl-T>73PASSWORD' to place 'REMOTE18' in columns 16-23 and 'PASSWORD' in columns 73-80. You must always specify two digits following a control-T. Thus column 6 would be specified as <cntrl-T>06.

The following tables list currently available customer manuals for HP 2000 Systems products. This list supersedes the list in the last issue of the **Communicator**.

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

Copies of manuals and updates can be obtained from your local Sales and Service office. The address and telephone number of the office nearest to you are listed in the back of all customer manuals.

Update packages are free of charge. If you require an update package only, send your request to:

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

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

A few words about documentation terms:

New	A new manual refers only to the first printing of a manual. When first printed, a manual is assigned a part number.
Revised	A revised manual is a printing of an existing manual which incorporates new and/or changed information in its contents. For example, a manual is revised when an update package is incorporated into the manual: the manual gets a new print date and the update package disappears. Note that a revision to a manual effectively obsoletes the previous version of the manual.
Update	An update package is a supplement to an existing manual which contains new and/or changed information. Updates are issued when information must get to customers, yet it is inappropriate to issue a revised manual. An update has no part number; it is automatically included when you order the manual with which it is associated.

PART NUMBER	MANUAL TITLE	PRICE*	DATE	UPDATE
02000-90048	20856A Timeshared BASIC/2000, Level E, Reference Manual	\$10.00	9/75	
02000-90049	20856A Timeshared BASIC/2000, Level E, System Operator's Manual	5.00	9/74	8/75
02000-90055	2000C/2000F IDF Author's Manual	8.50	1/73	8/74
02000-90073	20854A Timeshared BASIC/2000, Level F, Reference Manual	7.50	12/75	
02000-90074	20854A Timeshared BASIC/2000, Level F, System Operator's Manual	10.00	6/75	10/75
02000-90080	HP 2000E to HP 2000F Conversion Guide	1.00	4/76	*R
02116-9077	20392A HP BASIC Reference Manual	15.00	9/74	
19665-90001	2000/F to 2000/Access System Upgrade Kit and Conversion Program Manual	2.00	1/76	
20240-90001	Telecommunications Supervisory Package/2000 Manager's Manual	5.00	1/76	
20240-90002	Telecommunications Supervisory Package/2000 User's Manual	3.50	1/76	
20308-90001	Instructional Management Facility Proctor's Manual	7.00	9/74	
20308-90003	Instructional Management Facility System Manager's Reference Manual	5.00	10/74	
20309-90001	Instructional Dialogue Facility Proctor's Manual	10.00	9/74	
20309-90003	Instructional Dialogue Facility Course Developer's Manual	6.00	8/74	
20309-90005	Instructional Dialogue Facility Author's Pocket Guide	3.50	10/74	
20310-90001	HP MATH Teacher's Handbook	5.00	9/74	
20310-90005	HP MATH Proctor's Manual	5.00	9/74	
20310-90007	HP MATH Curriculum Guide	20.00	7/74	
20311-90001	Timeshared Graphics for Tektronix Terminals	7.00	8/74	
20311-90003	Timeshared Graphics Plotting Package	5.00	8/74	
20352-90001	Educational Budget and Accounting System — System Overview	5.00	6/74	
20352-90002	Educational Budget and Accounting System Reference Manual	10.00	3/75	4/76
20352-90003	Educational Budget and Accounting System — Technical Manual	75.00	3/75	
20353-90001	Educational Payroll System — System Overview	3.50	10/74	
22687-90001	HP 2000/Access BASIC Reference Manual	10.00	9/75	
22687-90005	HP 2000 Access Operator's Manual	10.00	9/75	10/75
22687-90009	Learning Timeshare BASIC	3.50	5/75	
22690-90001	Instructional Management Facility for HP 2000 Access Proctor's Manual	6.50	9/75	
22690-90002	Instructional Management Facility for HP 2000 Access System Manager's Reference Manual	4.50	9/75	
22691-90001	Instructional Dialogue Facility for HP 2000 Access Proctor's Manual	6.00	9/75	
22691-90002	Instructional Dialogue Facility for HP 2000 Access Course Developers' Manual	5.00	9/75	
22691-90003	Instructional Dialogue Facility for HP 2000 Access Author's Manual	13.00	9/75	
22691-90004	Instructional Dialogue Facility for HP 2000 Access Author's Pocket Guide	3.00	9/75	
22692-90001	Course Writing Facility Reference Manual	16.50	12/75	
22693-90001	HP MATH for HP 2000 Access Teacher's Handbook	5.50	7/75	
22693-90002	HP MATH for HP 2000 Access Proctor's Manual	6.50	7/75	
22693-90003	HP MATH for HP 2000 Access Curriculum Guide	17.50	7/75	
22700-90001	FCOPY/2000 Reference Manual	4.50	1/76	
24383-90001	Course Writing Facility	15.00	5/74	
24384-90001	College Information System — System Overview	5.00	6/74	
24384-90003	College Information System Reference Manual	19.00	9/75	
24384-90005	College Information System Technical Manual	95.00	5/75	
24387-90001	Basic Analysis and Mapping Program Manual	18.00	6/74	5/75
24387-90002	Basic Analysis and Mapping Program Pocket Guide	1.00	6/74	
5951-1352	The Librarian	1.00	1/71	
5951-1353	Special Purpose Magnetic Tape Loader/CAI English	1.00	1/71	
5951-1381	DOS-M/2000C Timeshared BASIC File Handler	1.00	5/71	
5952-4490	20856A Timeshared BASIC/2000, Level E, Pocket Guide	0.15	10/74	
5952-4491	20854A Timeshared BASIC/2000, Level F, Pocket Guide	0.15	8/75	

*R = Revised

*N = New

*Prices listed are subject to change without notice.

The schedule for customer training courses on General Systems Division Products is outlined below and in the HP 3000 section of this publication. Included here are 2000 Access courses for the period, May through August 1976

GENERAL SYSTEMS DIVISION COURSE SCHEDULE

May – August 1976

Course Dates and Training Center Location

COURSE NUMBER	COURSE TITLE	LENGTH	PRICE	GENERAL SYSTEMS	EASTERN TRAINING
				SANTA CLARA	CENTER – ROCKVILLE
22973A	2000 Access, Data Entry, File Management and RJE	5 days	\$500	6/14/76 8/23/76	

Registration

Requests for enrollment in any of the above courses should be made through your local HP Sales Office. Your Sales Representative will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the training course, time of class, location and accommodations reserved.

Accommodations

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

Cancellations

In the event you are unable to attend a class for which you are registered, please notify the Training Center Registrar immediately in order that we may offer your seat to another student. To avoid paying for a reservation which you do not use, we must receive notification of your cancellation no later than two weeks before the class begins.

EASTERN TRAINING CENTER

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

GENERAL SYSTEMS DIV. TRAINING CENTER

Hewlett-Packard
5303 Stevens Creek Blvd.
Santa Clara, Calif. 95050
(408) 249-7020

TIP ON USING EDIT/3000

Dick Sleight
HP General Systems

When you sign on to the HP 3000 with a terminal connected to a port that has a smaller configured record width than your terminal, you can use:

- The formal designator "EDITOUT" to change the output size
- "EDITIN" to change the input size.

The example below illustrates the use of an 80-character device on a configured 72-character port. Remember that the FILE command must go before the EDITOR command.

```
:FILE EDITOUT=$STDLIST;REC=80
:FILE EDITIN=$STDINX;REC=80
:EDITOR
```

```
HP32001A.4.03 EDIT/3000 SAT, APR 3, 1976, 12:01 PM
/SET LENGTH=80, RIGHT=80
/ADD
```

```
1 1234567890123456789012345678901234567890123456789012345678901234567890
2          2          3          4          5          6          7          8
```

With a 132-character terminal you can obtain an editor listing formatted the same as the offline option without folding at the configured width. For example, :FILE EDITOUT=\$STDLIST;REC=132.

RPG IMPROVEMENT FOR IBM S/3 CONVERSION

Joe Dietzgen
HP General Systems

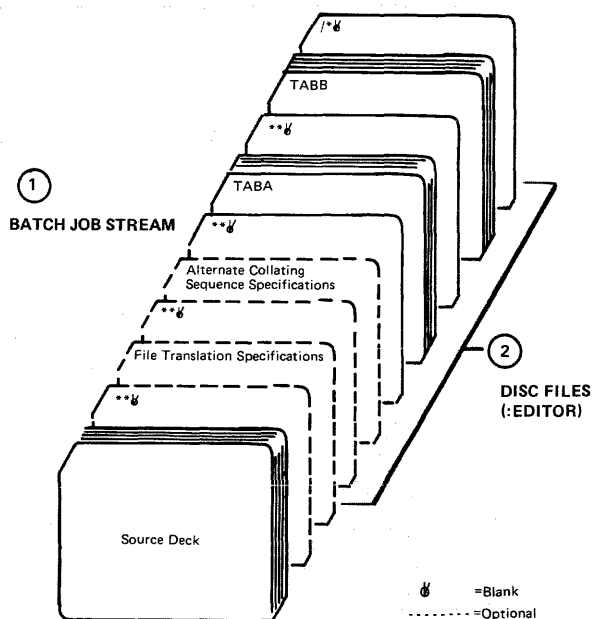
The RPG/3000 Compiler has been expanded to handle S/3 job streams for special records. (See the diagram below.)

RPG special records include:

1. File translation specifications
2. Alternating collating sequence specifications
3. Compile-time tables and arrays

RPG/3000 will now compile these records directly in the job stream. This method is used by the S/3.

RPG/3000 has an additional benefit for new programs. Any of these special records may be stored on disc files. This makes it easy to generate new Compile-Time tables and arrays and modify them with the :EDITOR. These disc records are accessed with A card disc file specifications. The A specifications are placed in the source file and are described in the RPG/3000 Compiler Reference Manual (part number 32104-90001).



RPG RECORDS

NEW RELEASE OF FORTRAN/3000

Bob Olson
HP General Systems

FORTTRAN version B (HP 32102B.00.00) is FORTRAN version A with a number of significant new features. This article describes the new features and suggests the applications for which they might be useful. A description of the syntax of the enhancements can be found in the 3000 Software Update section of the **Communicator**. The FORTRAN manual (32102-90001) has been revised to reflect these new features and is available. (See the Bulletins section for details.) It is important to note that all programs which execute under Version A should compile and execute under Version B with no changes necessary.

The enhancements fall into two major classes. The first class includes such extensions to the FORTRAN language as the new double integer data type, the generic function capability and the system intrinsic statement. The second class consists of added compiler control options such as the built-in cross reference facility and the listing of code offsets of statements down the left hand side of the listing, as in SPL.

Except for double integer, the choice of enhancements was not easy. Our selections were based on two user surveys, and suggestions from users and HP field support people. We took the list thus generated and gave each possibility an "expected value" based on implementation cost versus value to our users. We then made those enhancements which had a high "expected value" and which we could fit into the time we had available. If our judgments were correct, this new version of FORTRAN will be of considerable help to many of you.

New features are described below.

DOUBLE INTEGER

A double integer is a 32-bit two's complement integer. The range of a double integer is -2147483648 to 2147483647. Double integers may be used anywhere a single integer may appear, including but not limited to DO loop parameters, intrinsic arguments, and mixed mode arithmetic. Use of a double integer does not preclude use of single word integers in any way.

For FORTRAN programs currently running on other computer systems, which require 32 bit integers, an aid has been implemented. The \$INTEGER*4 command will cause the compiler to treat explicit and implicit INTEGER types as 32-bit integers; in addition all decimal integer constants will be treated as 32-bit integers. It is still possible to force variables to be single integers, for the sake of efficiency. This should improve the transportability of FORTRAN programs between the 3000 and other computer systems.

BUILT-IN TRAP HANDLING

The trap handling mechanism allows a user to write a subroutine to which control is transferred if a trap condition is encountered. Upon exit from the user-written subroutine the user stack is automatically adjusted and control returns to the instruction following the one that activated the trap.

The following four classes of traps may be handled:

1. Arithmetic traps (excepting decimal arithmetic)
2. Various Compiler Library traps (Formatter, arithmetic)
3. System Intrinsic error traps
4. Control Y traps

Since the user-written subroutine is written in FORTRAN and is compiled and prepared with the user program, the trap subroutine can communicate with the rest of the program through COMMON like any other subroutine. Furthermore, if the trap is an error trap, means are provided by which programmatic error recovery may be attempted. This feature may be of particular value to the numerical analyst or any programmer who wishes to specify his own handling for error conditions. The interface to Control Y may be of significant value in interactive applications.

SECONDARY ENTRY POINTS

Secondary entry points allow a user to begin execution of a subprogram at different locations. The name specified in a FUNCTION or SUBROUTINE statement is the primary entry point and the names used in any ENTRY statements are secondary entry points. In each case the invocation of a subprogram results in the transfer of control to the first executable statement following the entry name (primary or secondary) by which the subprogram was invoked. If an entry is identified as a function, its type is determined by the same rules as those governing other typed variables. The types of different entries in a function subprogram need not all be the same. Parameters are associated by name only, and may appear in any order and number, independently of any other entry to the subroutine.

GENERIC FUNCTIONS

Generic function names simplify the referencing of intrinsic and basic external functions because the same function name may be used with more than one type of argument. The type of the result is determined by the type of the argument, not the function name used.

SYSTEM INTRINSIC STATEMENT

The file SPLINTR.PUB.SYS contains information concerning SPL attributes of a set of subprograms, such as FOPEN. The information about a specific subprogram includes such information as the number and type of parameters, whether

parameters are by value or reference, and whether the subprogram is SPL OPTION VARIABLE. Under FORTRAN B, the appearance of "SYSTEM INTRINSIC" followed by a list of system intrinsic names will result in the SPLINTR file being searched for the names.

Use of this facility provides three main conveniences over the usual way of accessing external subprograms.

The first feature is convenient access to SPL OPTION VARIABLE routines. For each such subprogram (such as FOPEN, WHO, FGETINFO) the list of actual parameters need not be complete and may not include the SPL OPTION VARIABLE bit map indicating which parameters are present. The bit map and all required dummy values are automatically provided by FORTRAN.

The second feature is the compiler's ability to recognize the value or reference attribute of a formal parameter and automatically generate the appropriate code to stack actual parameters for the call.

The third facility provided is automatic typing of any of the names which are function names. For example, SYSTEM INTRINSIC FOPEN, BINARY will result in FOPEN being typed as integer and BINARY typed as logical and so forth.

In the past, the use of system intrinsics from FORTRAN has caused difficulties. The use of value arguments where reference was required, and vice versa resulted in many hours of debugging. The new System Intrinsic statement should also eliminate most of the tedium previously involved in the use of OPTION VARIABLE intrinsics. Note that only *MPE* system intrinsics are declared in the SYSTEM INTRINSIC statements (*not* FORTRAN/3000 intrinsic functions).

PARAMETER STATEMENT

A PARAMETER statement allows a constant to be given a symbolic name. FORTRAN refers to the constant as a parametric value. All types are allowed, including character. The name may then appear anywhere a constant may appear, with a few exceptions. The type of the name is determined solely by the constant and not the initial letter of the name. This technique has several advantages. Frequently it is desirable for documentation and enhancement purposes.

- A symbolic name has more intrinsic meaning than a numeric value.
- Parametric values are useful in applications controlled by tables.

(When such applications are being developed it is usually necessary to alter the table sizes and organization many times. The use of parametric values reduces the number of statements which must be modified to accommodate such table alterations)

UNDEFINED VARIABLE DETECTION

During compilation, FORTRAN/3000 issues a warning if it determines that a variable has been used without being initialized.

Since FORTRAN must globally analyze the treatment of variables, it does not detect when variables are referenced prior to their initialization — only when variables are not defined anywhere in the program.

Undefined variable detection is particularly useful in locating typographical errors. As most FORTRAN users know, the detection of typing errors can greatly ease the debugging process.

\$CONTROL LOCATION

This compiler control command forces a listing of the address of the start of each statement, relative to the beginning of the code generated for the program unit currently being compiled.

\$CONTROL CROSSREF

This compiler control command lists a cross reference of your program.

Listed below are other new features included in FORTRAN B.

The \$CONTROL FILE command has been extended to accept a range of logical unit numbers.

A new command has been added to allow users to override much of the parameter checking done by the segmenter at prepare time.

A new command has been added to provide more detailed compilation statistics.

New USL files will now be closed with a save disposition at the conclusion of the compilation.

A FORMAT statement may contain an octal number in the range 0–377 which will be treated as being equivalent to a byte character. The primary function served is that of a carriage control character, especially where a particular number does not represent an available ASCII character.

We hope that these new features will enhance the usability of your systems and ease your FORTRAN programming tasks. A major factor in getting these enhancements implemented was your visible interest in having these capabilities. Your comments and suggestions are always welcome, as they help us define the weaknesses and strengths of our products.

Please send any comments and suggestions to the Planning Committee of the HP 3000 Users Group in care of:

Bill Gates
Longs Drugs
141 N. Civic Drive
Walnut Creek, Ca 94596

NEW FORTRAN/3000 MANUAL

Sandy Martensen
HP General Systems

HOW TO OBTAIN AN UPDATE TO A MANUAL

Ilene Birkwood
HP General Systems

If you would like a copy of the MPE update, or any other update listed in the documentation section (page 355) of the **Communicator**, just send your request to:

Software/Publications Distribution
 Hewlett-Packard Company
 5303 Stevens Creek Blvd.
 Santa Clara, Ca. 95050

Please give the name of the manual, its part number, and state clearly that you require the update, not the complete manual. There is no charge for manual updates.

For example, to order the MPE update send us a letter giving your name and address and request:

MPE Reference Manual
 Part number 32000-90002
 UPDATE ONLY

NEW MPE/3000 REFERENCE MANUAL UPDATE

Barbara Lewis
HP General Systems

A new update to the *MPE/3000 Reference Manual* is now available. The new update incorporates all previous updates and reflects the following changes to the software, which were described in the November 15, 1975 and April 1, 1976 issues of the **Communicator**.

- An addition to the FUPDATE intrinsic and FPOINT intrinsic.
- A replacement to the Run-Time messages.
- A new FCHECK code.
- The PAUSE intrinsic.

The update is free of charge and can be ordered by specifying manual part number 32000-90002, update only. (Refer to the article in this section on ordering Manual Updates.)

By ordering the *MPE/3000 Reference Manual* and this update, you will have the complete package.

A new version of the FORTRAN/3000 Reference Manual is now available. It documents the features available with the newly released version of FORTRAN, HP 32102B, which is described in the Software Update section of this **Communicator**.

The new features include:

- double integers
- a mechanism for trap handling
- secondary entry points to subroutines and functions
- generic functions
- SYSTEM INTRINSIC statement
- PARAMETER statement
- undefined variable detection
- character strings allowed in STOP or PAUSE statements
- \$CONTROL command options, CROSSREF and LOCATION

The revised FORTRAN/3000 Reference Manual is dated March, 1976, part number is 32102-90001 and the price is \$13.50.

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

Software updates described in this issue relate to the following products. (There are no updates to the MPE operating system in the MIT Tape dated 1616.)

Product	Number	Fix Update & Level	MIT Tape Date Code
BASIC/3000	32101A	03.03	1616
BASICOMP/3000	32103A	00.05	1616
COBOL/3000	32213B	01.06	1616
FORTRAN/3000	32102B	00.00	1616
RPG/3000	32104	02.03	1616

Where changes in documentation are indicated, updates to the appropriate manuals will be printed. This information is provided simply as a temporary measure.

BASIC/3000, HP 32101A.03.03**PROBLEMS CORRECTED IN BASIC/3000**

1. A string literal consisting of more than one non-printable character was listed incorrectly (although stored correctly) for ASCII codes exceeding 127 (e.g., '128).

OUTSTANDING PROBLEMS IN BASIC/3000

1. The interpreter aborts with a stack underflow when control-Y is typed in certain circumstances. This occurs most often when printing the FREQ table. The problem may also arise in some cases when INVOKing or using the ABORT, CALLS or FILES commands in BREAK-mode.

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

2. The statement "PRINT #F;END" fails for one-record files.
3. \$NULL is always treated as a binary file. This causes the ADVANCE and file LINPUT statements to terminate with an error rather than recognize an end-of-file. The UPDATE statement also terminates with the wrong message.
4. The command "RUN ,NOECHO,OUT=. ." suppresses prompting when the input device is interactive with the list device.
5. The statement "PRINT #F,R" fails to set the end-of-record at the beginning of a BASIC formatted file.
6. Lower-case characters are not recognized as format specifications in PRINT USING format string expressions.

BASICOMP/3000, HP 32103A.00.05**PROBLEMS CORRECTED IN BASICOMP/3000**

1. Incorrect values were printed for relational expressions in a PRINT statement (e.g., PRINT P>Q).
2. The type-LONG MIN and MAX operators caused a bounds-violation abort at run-time. This occurred when either (but not both) operand was an expression or when the type-LONG operand was a simple or subscripted variable and the other operand was a different data-type.
3. A string literal consisting of more than one non-printable character was listed incorrectly (although stored correctly) for ASCII codes exceeding 127 (e.g., '128).

4. The compile-time INIT option caused arrays to be initialized to one rather than to the undefined value.
5. In extremely rare circumstances, integer FOR-loops generated an incorrect branch instruction. This resulted in various errors (e.g., abnormal termination of an enclosing FOR-loop) or it had no adverse effect at all, depending on previous statements in the program.

OUTSTANDING PROBLEMS IN BASICOMP/3000

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

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

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

Work-around: Fully parenthesize expression.

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

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

4. \$NULL is always treated as a binary file. This causes the ADVANCE and file LINPUT statements to terminate with an error rather than recognizing an end-of-file. The UPDATE statement also terminates with the wrong message.
5. Incorrect code is generated for a FOR-loop which encloses both an ONEND statement with a destination within the loop and a GOTO statement with a destination outside the loop. This situation will cause spurious run-time aborts if an end-of-file is detected while inside the FOR-loop.

Work-around: Place a superfluous GOTO statement outside the FOR-loop with a destination inside the loop. The GOTO statement itself is not intended to be executed.
6. Lower-case characters are not recognized as format specifications in PRINT USING format string expressions.

COBOL/3000, HP 32213B.01.06**PROBLEMS CORRECTED IN COBOL/3000**

1. DECIMAL-POINT IS COMMA caused fatal error in compilation when used in DYNAMIC subprograms.
2. DECIMAL-POINT IS COMMA caused incorrect code when using numeric literals.

FORTRAN/3000, HP 32102B**The following errors in version A have been corrected:**

1. Appearance of a subroutine call with no parameters followed by a call with parameters was not detected by the compiler. At code generation time the compiler attempted to process the non-existent parameter list of the first call. This resulted in the compiler aborting with a bounds violation.
2. Very complex branch statements (logical IFs) can overflow an internal table, resulting in incorrect branches around embedded constants. For the time being, the compiler will generate a warning message whenever it detects the condition. A complete fix probably requires a redesign of the affected mechanism. A rule of thumb for avoiding this problem is that if you have a hard time understanding the conditional expression, the compiler may also be confused.
3. On occasion, the table of unallocated constants will become full in the middle of generating code to evaluate a conditional expression for a logical IF statement. If more than %77 words of constants are emitted, the indirect cell for the previous branch is not correct. This problem is related to problem 2 mentioned above and could be experienced in any situation where very little code is generated relative to the number of constants generated. The problem can usually be worked around by doing anything which will move the conditional branches forward or backward in the code. For example, insertion of a dummy assignment or reshuffling the conditional expression might avoid the problem. Until a more satisfactory solution to the problem can be found, the compiler will emit a warning message whenever it detects the condition.
4. Interactions between logical IF statements and action conditions in I/O statements sometimes resulted in the action label interfering with the placing of an indirect cell for the logical IF statement.
5. Redundant initialization of a data item in a DATA statement could cause bad USL information. A new error message was added to catch this user error condition.
6. Large statement functions, in either size or quantity, sometimes resulted in bad initialization code.
7. Byte address calculations involving substring designators with constants used an integer instruction, resulting in integer overflow when the stack crosses 32K bytes.

Known Problems with Version B.00.00

None, except those now resolved with a warning message.

Known differences between versions A and B:

1. All programs which execute under version A should execute under version B with no modifications. Any programs which fail to do so should be reported to General Systems Division.
2. Listed below is a partial list of enhancements and improvements. Refer to the revised FORTRAN/3000 manual for details.

A. Double Integer

A double integer is a 32-bit two's complement integer. The range of a double integer is -2147483648 to 2147483647. Double integers may be used anywhere a single integer may appear, including but not limited to DO loop parameters, intrinsic arguments, and mixed mode arithmetic. Use of a double integer does not preclude use of single word integers in any way.

Constants may be represented in decimal, octal, ASCII, and composite formats just as single word integers may be, except that they must all end with "J". For example, %30J.

Double integers may appear in any specification statement in which single integers may appear. A name is specified to be of type double integer by using the type attribute "INTEGER*4" wherever an integer type attribute is legal. The attribute "INTEGER*2" may be used as well as "INTEGER" for single word integers. The default integer is a single integer.

The following intrinsic and basic external functions are available. They are exact counterparts of the similarly named single integer functions.

JABS, JINT, JDINT, JMOD, FLOATJ, JFIX, JSIGN, JDIM, AJMAX0, JMAX0, JMAX1, AJMIN0, JMIN0, JMIN1, JNUM.

IJINT — Double integer to integer conversion

JiINT — Single integer to double integer conversion.

For FORTRAN programs currently running on other systems which require 32-bit integers an aid has been implemented. The \$INTEGER*4 compiler command will cause the compiler to treat explicit and implicit INTEGER types as 32-bit integers and decimal integer constants as 32-bit constants. The INTEGER*2 type is still available for explicit typing of variables as single

integers. The following intrinsics will accept and/or return 32-bit integers in place of their normal 16-bit integer values: BOOL, MOD, IDIM, IABLS, IFIX, INT, FLOAT, MAX0, MAX1, MIN0, MIN1, AMAX0, AMIN0, ISIGN, IDINT.

B. Trap Handling

The trap handling mechanism allows a user to write a subroutine to which control is transferred if a trap condition is encountered. Upon exit from the user provided subroutine the user stack is automatically adjusted and control returns to the instruction following the one that activated the trap. The following types of traps may be handled:

1. Real divide by zero
2. Real overflow
3. Real underflow
4. Double precision divide by zero
5. Double precision overflow
6. Double precision underflow
7. Integer*2 divide by zero
8. Integer*2 overflow
9. Integer*4 divide by zero
10. Integer*4 overflow
11. System intrinsic errors
12. Basic external function errors
13. FORTRAN internal function errors
14. FORTRAN Formatter errors
15. Plotter errors
16. Control Y traps

Since the user-written subroutine is written in FORTRAN and is compiled and prepared with the user program, the trap subroutine can communicate with the rest of the program through COMMON like any other subroutine. Furthermore, if the trap is an error trap, means are provided by which programmatic error recovery may be attempted. This feature may be of particular value to the numerical analyst or any programmer who wishes to specify his own handling for error conditions. The interface to Control Y may be of significant value in interactive applications.

The built-in trap handling mechanism is rather too complex to be described in a useful manner in this note. If your application requires this facility, you should obtain the revised manual (32102-90001).

C. Secondary Entry Points

Multiple entry points allow a user to begin execution of a subprogram at different locations. The name specified in a FUNCTION or SUBROUTINE statement is the primary entry point and the names used in ENTRY statements are secondary entry points. In each case the invocation of a subprogram results in the transfer of control to the first executable statement following the entry name (primary or secondary) by which the subprogram was invoked.

The form of an ENTRY statement is

```
ENTRY entryname [(parameter[,parameter...])]
```

An ENTRY in a SUBROUTINE is a subroutine entry and an ENTRY in a FUNCTION is a function entry. One or more ENTRY statements are allowed. All entry points are reentrant. An entry statement may appear anywhere except in the range of a DO statement.

If an entry is identified as a function, its type is determined by the same rules as those governing other typed variables. The types of different entries in a function subprogram need not all be the same. The entries of the same type all share the same return space and if entry is made through an entry name of a given type then all other entry names of the same type may be referenced. Any attempt to use an entry name which is of a different type than that by which entry was made as anything other than a function call will result in the program aborting with a bounds violation. A type attribute preceding the word ENTRY in an entry statement expressly prohibited.

Only those parameters which are associated with the entry through which entry was made are defined. Any attempt to access formal parameters not part of said entry will result in the program aborting with a bounds violation.

Parameters are associated by name only, and may appear in any order and number, independently of any other entry to the subroutine. For example,

```
INTEGER FUNCTION IFN(A,B)
  LOGICAL LFN
  ----
  ----
  ENTRY LFN(L,A)
  ----
  ----
  ENTRY AFN(L,B,A)
  ----
```

LFN will be defined as a logical and AFN will be a real. A, B, and L will refer to the same entity even though on entry they differ in type and order.

D. Generic Functions

Generic function names simplify the referencing of intrinsic and basic external functions because the same function name may be used with more than one type of argument. The type of the result is determined by the type of the argument, not the function name used. For example,

```
DOUBLE PRECISION D
D = TAN(D)
```

generates exactly the same result as

```
DOUBLE PRECISION D,DTAN
D = DTAN(D)
```

There are fairly specific rules about what happens if a generic name is redefined. Those rules can be found in the updated manual for those interested. The list of generic names is:

ABS, INT, JINT, REAL, MOD, SIGN, DIM, MAX, MIN, SQRT, EXP, LOG, SIN, COS, TAN, ATAN, SINH, COSH, TANH

In general, the correspondence between the generic name for a class of functions and the specific names of the functions in that class are obvious.

E. System Intrinsic Statement

The file SPLINTR.PUB.SYS contains information concerning SPL attributes of a set of subprograms, such as FOPEN. The information about a specific subprogram includes such information as the number and type of parameters, whether parameters are by value or reference, and whether the subprogram is SPL OPTION VARIABLE. Under FORTRAN B, the appearance of "SYSTEM INTRINSIC" followed by a list of system intrinsic names will result in the SPLINTR file being searched for the names.

Use of this facility provides three main conveniences over the usual way of accessing external subprograms. The first feature is convenient access to SPL OPTION VARIABLE routines. For each such subprogram (such as FOPEN, WHO, FGETINFO) the list of actual parameters need not be complete and may not include the SPL OPTION VARIABLE bit map which indicates which parameters are present. The bit map and all required dummy values are automatically provided by FORTRAN.

The second feature is the compiler's ability to recognize the value or reference attribute of a formal parameter and automatically generate the appropriate code to stack actual parameters for the call.

The third facility provided is automatic typing of any of the names which are function names. For example, SYSTEM INTRINSIC FOPEN, BINARY will result in FOPEN being typed as integer and BINARY typed as logical and so forth.

In the past, the use of system intrinsics from FORTRAN has caused difficulties. The use of value arguments where reference was required, and vice versa resulted in many hours of debugging. The new System Intrinsic statement should also eliminate most of the tedium previously involved in the use of OPTION VARIABLE intrinsics. Note that only *MPE* system intrinsics are declared in the SYSTEM INTRINSIC statements (*not* FORTRAN/3000 intrinsic functions).

F. Parameter Statement

A PARAMETER statement allows a constant to be given a symbolic name. FORTRAN refers to the constant as a parametric value. All types are allowed, including character. The name may then appear anywhere a constant may appear, with a few exceptions. The type of the name is determined solely by the constant and not the initial letter of the name. This technique has several advantages:

- Frequently it is desirable for documentation and enhancement purposes.
- A symbolic name has more intrinsic meaning than a numeric value.
- Parametric values are useful in applications controlled by tables. (When such applications are being developed it is usually necessary to alter the table sizes and organization many times. The use of parametric values reduces the number of statements which must be modified to accommodate such table alterations.)

The form of the statement is

```
PARAMETER name=constant [,name=constant...]
```

The name may not be used as part of another constant except as the real or imaginary part of a complex constant. A parameter statement must appear before DATA statements, statement functions and executable statements.

Examples:

```
PARAMETER !=%3615D           double precision
PARAMETER N=3.2E+5, S=1       real, integer
```

G. Carriage Control Character

A FORMAT statement may contain an octal number in the range 0–377 which will be treated as being equivalent to a byte character. The primary function served is that of a carriage control character, especially where a particular number does not represent an available or representable ASCII character. The octal number must be distinguished by a preceding percent sign and a trailing "C".

Example:

```
FORMAT(%306C,5HHELLO)
(spaces 1/4 page before printing HELLO)
```

H. Undefined Variable Detection

A warning message is emitted if a variable appears in one or more of the following.

1. Right hand side of an assignment statement.

2. As a value parameter to a subprogram.
3. As a list element in a WRITE or DISPLAY statement.

and does not appear in one or more of the following.

1. Left hand side of an assignment statement.
2. As a list element in a READ statement.
3. In a DATA statement.
4. As a formal parameter.
5. As a reference parameter to a subprogram.
6. In an EQUIVALENCE statement.
7. In a COMMON statement.

Since FORTRAN must globally analyze the treatment of variables, it does not detect when variables are referenced prior to their initialization — only when variables are not defined anywhere in the program.

Undefined variable detection is particularly useful in locating typographical errors. As most FORTRAN users know, the detection of typing errors can greatly ease the debugging process.

I. Character Strings in STOP or PAUSE

The allowable parameter following a STOP or PAUSE statement may be a character string enclosed by quote marks or apostrophes, up to 254 characters.

J. \$CONTROL CROSSREF

The option to have the compiler list a cross reference of your program is invoked by this compiler control command. At the end of each program unit a listing of all symbolic names in alphabetic order and all statement labels will be made with all sequence numbers where the name or label appear immediately following each name or label. In addition, by using the command

```
$CONTROL CROSSREF ALL
```

it is possible to have a cross reference of all names which have global significance, such as common block names, subroutine names, function names and program names. Such a cross reference can be invaluable in program maintenance, debugging and program conversion.

K. \$CONTROL LOCATION

This compiler control command forces a listing of the address of the start of each statement, relative to the beginning of the code generated for the program unit currently being compiled. The addresses will be listed down the left hand edge of the page, much as SPL does. This option, in combination with the CODE option and the PMAP generated by the Segmenter when you prepare your program, can be a very powerful debugging tool. We used this option heavily in debugging the code generated by the new FORTRAN compiler.

L. Other New Features in FORTRAN B

The \$CONTROL FILE command has been extended to accept a range of logical unit numbers. i.e., \$CONTROL FILE=5-10.

A new command has been added to allow users to override much of the parameter checking done by the Segmenter at prepare time. \$CONTROL CHECK=n. Refer to FORTRAN manual (32102-90001) for details on parameter significance.

A new command has been added to provide more detailed compilation statistics. \$CONTROL STAT/NOSTAT.

New USL files will now be closed with a save disposition at the conclusion of the compilation.

RPG/3000 HP 32104.02.03

PROBLEMS CORRECTED IN RPG/3000

1. A move of a 256 byte field to a 256 byte field was a null operation.
2. An invalid entry error in the USL file occurred while preparing an RPG generated program.
3. If RLABELs referred to an already defined field a code generation error occurred.
4. If PARM was not preceded by an EXIT the compiler aborted with a bounds violation.
5. If input to an RPG generated program was spooled from a card reader/punch/interpreter an invalid control operation was issued.

ENHANCEMENTS TO RPG/3000

1. If IMAGE files are defined in an RPG program the IMAGE intrinsics are loaded at program load time instead of dynamically at execution time. This will greatly decrease the time necessary to get the program running.
2. Multiple RSAM files in an RPG program may now fully share the same RSAM file.

The user may specify several files to RPG, all with different file names and have all reference the same file using the same file number.

The user simply inserts a file continuation record after the RSAM file specification he wishes shared.

The continuation record format is:

COI	CONTENTS
6	"F"
53	"K"
54-59	"DSNAME"
60-67	RSAM file name

FOR EXAMPLE

```

F FILENAMEIP F      80   A I    10 DISC
F

```

K DSNAMERSAMNAME

This declares that the file filename is really the RSAM file RSAMNAME.

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

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

Copies of manuals and updates can be obtained from your local Sales and Service office. The address and telephone number of the office nearest to you are listed in the back of all customer manuals.

Update packages are free of charge. If you require an update package only, send your request to:

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

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

A few words about documentation terms:

- | | |
|---------|---|
| New | A new manual refers only to the first printing of a manual. When first printed, a manual is assigned a part number. |
| Revised | A revised manual is a printing of an existing manual which incorporates new and/or changed information in its contents. For example, a manual is revised when an update package is incorporated into the manual: the manual gets a new print date and the update package disappears. Note that a revision to a manual effectively obsoletes the previous version of the manual. |
| Update | An update package is a supplement to an existing manual which contains new and/or changed information. Updates are issued when information must get to customers, yet it is inappropriate to issue a revised manual. An update has no part number, it is automatically included when you order the manual with which it is associated. |

MPE/3000 MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
03000-90096	Multiprogramming Executive General Information Manual	\$ 4.00	11/73	
03000-90126	HP 3000 Software Pocket Guide	3.50	7/75	
32000-90002	32000C MPE/3000 Reference Manual	19.50	1/75	3/76
32000-90004	32000C MPE/3000 Console Operator's Guide	7.00	1/75	10/75
32000-90006	32000C MPE/3000 System Manager/System Supervisor Manual	13.00	10/75	
32000-90008	MPE/3000 Operating System, System Utilities Manual	3.00	10/75	

LANGUAGE MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
03000-90002	SPL/3000 Reference Manual	\$ 7.50	11/73	
03000-90003	SPL/3000 Textbook	13.00	11/73	3/75
03000-90008	BASIC/3000 Interpreter Reference Manual	10.00	7/75	
03000-90025	BASIC for Beginners	5.50	11/72	
03000-90047	Cross Assembler for 2100 Computers Reference and Application Manual	17.00	3/75	
03000-90050	BASIC/3000 Interpreter Pocket Guide	2.50	9/74	
32102-90001	FORTRAN/3000 Reference Manual	13.50	3/76	*R
32103-90001	BASIC/3000 Compiler Reference Manual	3.50	11/74	
32104-90001	RPG/3000 Compiler Reference and Application Manual	20.00	2/75	
32104-90003	RPG Listing Analyzer	0.50	4/75	
32213-90001	COBOL/3000 Reference Manual	12.50	7/75	10/75

ADDITIONAL MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
03000-90009	HP 3000 Compiler Library Reference Manual	\$16.50	2/76	
03000-90010	HP 3000 Scientific Library Reference Manual	7.00	7/75	
03000-90011	STAR/3000 (Statistical Analysis Routines) Reference Manual	5.50	11/72	
03000-90012	EDIT/3000 Reference Manual	7.50	8/75	
03000-90015	HP 3000 Symbol Trace Reference Manual	7.50	10/75	
03000-90019	HP 3000 Computer Systems Reference Manual	14.00	9/73	
03000-90064	FCOPY/3000 Reference Manual	6.00	7/75	
03000-90107	HP 3000 Cross Loader for HP 2100 Computers	11.00	10/74	
03000-90121	Using the HP 3000: A Guide for the Terminal User	7.50	6/75	
30130-90001	2780/3780 Emulator Subsystem Reference and Application Manual	10.00	12/74	2/76
30300-90002	Programmable Controller Reference and Application Manual	12.00	2/75	4/76
32104-90004	IBM SYSTEM/3 to HP 3000 Conversion Guide	7.50	12/75	
32215-90001	IMAGE/3000 Reference Manual	7.00	3/76	
32216-90001	QUERY/3000 Reference Manual	7.00	3/76	
32900-90001	Student Information System Reference Manual	18.00	9/74	
32900-90002	Student Information System — System Overview	7.00	9/74	
32900-90005	Student Information System — Technical Manual	18.50	3/75	
36995-90013	IBM 1130/1800 to HP 3000 FORTRAN Conversion Guide	6.00	2/75	5/75
32214-90001	Sort/3000 Reference Manual	6.50	4/75	
30301-90002	Real-Time Programmable Controller Reference Manual	9.50	2/75	
32901-90001	Student Assignment System Reference Manual	10.00	7/75	
32901-90005	Student Assignment System Technical Manual	13.00	7/75	

*R = Revised

The schedule for customer training courses on General Systems Division products is outlined below, and in the 2000 Access section of this publication. Included here are HP 3000 software courses offered in the U.S. and in Europe, for the period May through August, 1976. You can also obtain a copy of the schedule from your local HP sales office. A European course schedule is available through the sales offices in Europe; a U.S. schedule through U.S. sales office.

Registration

Requests for enrollment in any of the courses should be made through your local HP Sales Office. Your Sales Representative will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the training course, time of class, location and accommodations reserved.

GENERAL SYSTEMS DIVISION COURSE SCHEDULE (U.S.)

May — August 1976

Course Dates and Training Center Location

NUMBER	COURSE TITLE	LENGTH	PRICE	GENERAL SYSTEMS SANTA CLARA	EASTERN TRAINING CENTER — ROCKVILLE
22956A	3000 IMAGE	5 days	\$500	5/3/76 6/7/76 7/12/76 8/16/76	5/17/76 7/26/76
22957A	3000 COBOL, Audio Self Study	30 hrs.	325	These courses can be ordered using the Direct Mail Order form in the back of the Communicator .	
22958A	3000 BASIC, Audio Self Study	30 hrs.	325		
22962A	3000 Commercial/Business User	5 days	500		5/3/76
22964A	3000 System Management	3 days	300		5/10/76
22975A	System 3 Conversion Seminar	2 days	200	6/14/76 8/23/76	
*22801A	3000, A Comprehensive Introduction (Replaces 22962A)	5 days	500	5/24/76 6/28/76 8/2/76 8/30/76	6/7/76 7/12/76 8/2/76 8/16/76
*22802A	3000 System Management and Operation (Replaces 22964A)	4 days	400	6/1/76 7/6/76 8/9/76	6/14/76 7/19/76 8/9/76 8/23/76

**New Courses: Course 22801A, HP 3000: A Comprehensive Introduction is a new offering. It is a combination and redesign of the 3000 Commercial and 3000 Scientific courses. The curriculum now requires less specific programming language skills on the part of the student. Course 22802A, HP 3000: System Management and Operation replaces 22964A (3000 System Management); the new class adds additional material and extends the training from 3 to 4 days.*

Accommodations

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

Cancellations

In the event you are unable to attend a class for which you are registered, please notify the Training Center Registrar immediately in order that we may offer your seat to another student. To avoid paying for a reservation which you do not use, we must receive notification of your cancellation no later than two weeks before the class begins.

GENERAL SYSTEMS DIVISION COURSE SCHEDULE (EUROPE)

May — August 1976

COURSE NUMBER	COURSE TITLE	LENGTH	BOBLINGEN (GERMAN)	WINNERSH (ENGLISH)	ORSAY (FRENCH)	MILAN (ITALIAN)	STOCKHOLM (ENGLISH)
22962A	3000 Commercial/ Business User	5 days	8/9	5/17			5/10 8/30
22963A	3000 Scientific/ Engineering User	5 days	6/21		6/28		6/14
22964A	3000 System Management	3 days			5/22 6/21	6/30	5/17

Training Center Addresses

Santa Clara
5303 Stevens Creek Blvd.
Santa Clara, California 95050
(408) 249-7020

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

Boblingen
Kundenschulung
Herrenbergerstrasse 110
D-7030 Böblingen, Wurttemberg
Tel: (07031) 667-1
Telex: 07265739
Cable: HEPAG

Winnersh
King Street Lane
GB-Winnersh, Wokingham
Berks RG11 5 AR
Tel: Wokingham 784774
Cable: Hewpie London
Telex: 847178 9

Grenoble
5, avenue Raymond-Chanas
38320 Eybens
Tel: (76) 25-81-41
Telex: 980124

Milan
Via Amerigo Vespucci, 2
1-20124 Milan
Tel: (2) 62 51
Cable: HEWPACKIT Milano
Telex: 32046

Stockholm
Enighetsvägen 1-3, Fack
S-161 20 Bromma 20
Tel: (08) 730 05 50
Cable: MEASUREMENTS
Stockholm Telex: 10721



introducing HP 3000 series II computer systems

In May, 1976, Hewlett-Packard is announcing a new generation of HP 3000 computers, the HP 3000 Series II. This new family of systems embodies the same architecture as the original HP 3000, and (with a few exceptions) maintains software capability with its predecessor. The new Series II contains a new higher performance central processor, new multiplexer and selector channels and a new fault-control semiconductor memory system which is expandable to 512K bytes of main memory.

All memory on the new Series II is organized into 21-bit quantities where five bits are identified as checking bits. The memory system is able to detect and correct all single-bit memory errors which may occur during memory operation. Any errors detected are logged in a logging RAM. The contents of this RAM are periodically written to a disc file by MPE. The service engineer accesses this disc file at preventive maintenance time and replaces any faulty memory chips.

This new family consists of three models:

Model 5 — a starter system with 128K bytes of main memory, a 15M byte disc drive, 1600 BPI tape unit, and a 2640A System console.

Model 7 — a data base-oriented system with 192K bytes of main memory, two 47M byte disc drives, 1600 BPI tape unit, 2640A system console, COBOL, RPG, IMAGE and QUERY.

Model 9 — a high performance system with 320K bytes of main memory, two 47M byte disc drives, a 1600 BPI mag tape unit, a 2640A system console, FORTRAN, BASIC interpreter and compiler, COBOL, RPG, IMAGE, QUERY, the Scientific Library, and a new Data Entry Library (a set of programs and subroutines which simplify forms design, data entry, and data editing using HP 2640A and 2644A terminals).

All models contain the new MPE-II operating system, EDIT, SORT, and TRACE utilities, and the SPL compiler. All models also contain decimal firmware instructions and extended precision floating point instructions, which on the Series II are represented as 64-bit quantities. (Extended precision numbers on the original HP 3000 are represented as 48-bit quantities.) Line printers, card equipment, additional disc drives, tape drives, and terminals can be added to any of the models. Main memory in the Models 5 and 7 is expandable to 256K bytes; the model 9 is expandable to 512K.

All existing HP 3000's can be upgraded to a Series II configuration that contains at least 256K bytes of main memory. Upgraded configurations are fully expandable to 512K bytes.

A COMPLETE RANGE OF SUPPORT SERVICES

TO INSURE THE SUCCESS OF YOUR HP 3000 INSTALLATION

Concurrent with the Series II announcement, HP will be consolidating its support services to provide an efficient, professional means of obtaining hardware and software maintenance, training and consulting to complement your HP 3000 system.

Shown below is a summary of our services which, subject to availability in your area, will be available to all HP 3000 customers beginning in July 1976. While some of the services have been offered previously, others such as on-site training and consulting are new.

The comprehensive selection of support services offered can be classified into three general areas:

Site preparation and installation. This service, which is included in the price of the system, assures you of successful operation by providing assistance in preparing your site, and by installing your system.

Professional training and consulting for your staff. A complete selection of training courses and consulting packages may be purchased to allow you to tailor a training program to your staff's needs.

Maintenance Program. A complete system maintenance program contracted for on a yearly basis provides:

- Hardware maintenance
- Software maintenance
- Reference manual updates
- Information newsletter

Good support depends on competent people. The following are brief descriptions of the Hewlett-Packard team that will provide support for your site.

- **Sales Representative** — The person responsible for handling your account on a continuing basis. When you need new solutions, additional system capability, extra training courses, technical consulting, or if you have any problems or questions — this is the person you contact.
- **Systems Engineer (S.E.)** — A technical specialist who works with the Sales Representative to insure that your application runs smoothly on the HP 3000. The S.E. has had extensive training on the subsystems and operating system, and can provide on-site training and consulting on their use. Your S.E. can also provide assistance to resolve problems you may be having in using your system.

- **Customer Engineer (C.E.)** — Your service representative. Upon purchase of a system maintenance contract, Hewlett-Packard assigns an individual C.E. to be responsible for your system. The C.E. can help you with any service problem — carries a complete set of tools and spare parts and has been trained to quickly troubleshoot and maintain HP 3000 systems. The C.E. will also call in any other help that may be required to keep your system up and running. Your C.E. is backed by an HP 3000 system Product Specialist — a behind-the-scenes technical expert who provides hardware and operating system software backup for your HP Customer Engineer, and has extensive experience with the HP 3000 operating system. The Product Specialist also provides local assistance in analyzing possible problems.

The services that your Hewlett-Packard support team provides are described in the following pages.

PREPARING YOUR SITE AND INSTALLING YOUR HP 3000 SYSTEM

SITE PREPARATION CONSULTING

After an order has been placed for an HP 3000 Computer System, your Sales Representative will make arrangements for the HP Customer Engineer who will be responsible for maintaining your system, to visit your site and consult with you on matters related to the environment of the system. The first visit occupies approximately four hours, exclusive of travel. The Customer Engineer evaluates the site with relation to the specifications contained in the HP 3000 Site Preparation Manual, and makes specific recommendations about any changes that are required. The Customer Engineer will visit your site a second time prior to system installation, to insure that the site is properly prepared. These visits are included in the purchase price of an HP 3000 System.



During his first visit, the engineer:

- Provides you a copy of the HP 3000 Site Preparation Manual, publication number 30000-90016. This information helps you and your company's Facilities Engineer make preparations for installation, well in advance of actual delivery of the system.
- Calculates the heat output of the proposed system, to advise whether adequate air-conditioning equipment is available to keep the system within heat, humidity and dust limits.
- Checks to determine if adequate power facilities exist and that they conform to noise and neutral-to-ground impedance requirements; advises you on requirements for isolation transformers.
- Verifies the general suitability of the proposed system environment with respect to space, lighting, storage space, access to peripherals, etc.
- Checks the means by which the system is to be installed. For example, he makes sure that elevators are adequate to support size and weight of equipment.
- Advises you of your responsibilities in preparing for the installation.

SYSTEM INSTALLATION

Installation service, which is included in the purchase price of an HP 3000 System, is directed by your local HP Customer Engineering group.

An HP Customer Engineer will install your HP 3000 System and insure that it is operating according to specifications. Installation service consists of the following:

- Supervising unpacking and taking inventory
- Installing equipment as defined in the sales order (except terminals)
- Turning power on
- Running all diagnostics successfully
- Running verification program. (This is the same verification procedure used by the manufacturing division at the time the system was integrated)
- Familiarizing the operator with the system (power on, daily maintenance, emergency procedures)
- Loading and verifying all ordered software
- Showing you how to obtain the HP services you need.

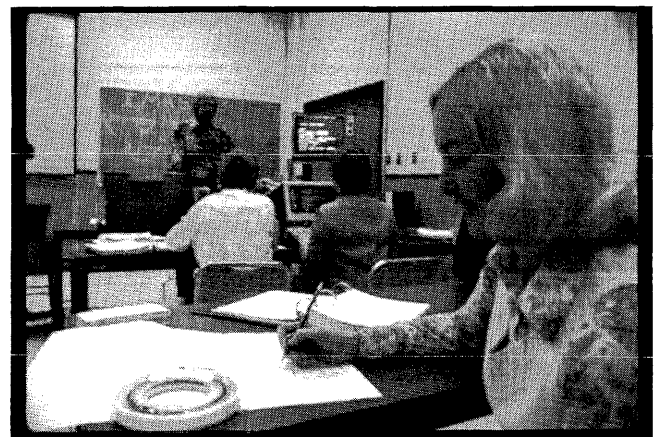
The following items which are **not** included in the system installation service, can be provided at your request, on a time and materials basis:

- Reconfiguring new systems
- Set-up of terminals (not including extensive cabling)

Your responsibilities:

- Physically moving the system from your delivery dock to your computer room
- Unpacking and taking inventory (under the supervision of the HP Customer Engineer)
- Connecting the system to the electrical supply
- Acquiring and installing data communications equipment (modems, telephone lines, etc.)
- All site modifications identified during the site preparation consulting visit.

Professional Training and Consulting for Your Staff



Hewlett-Packard offers a variety of training courses and consulting packages to get you started successfully using your HP 3000 System, and to provide on-going support as you develop new applications. A special module called the "HP 3000 Start-up Package" includes two fundamental training courses, plus three days of on-site HP 3000 Installation Start-Up Consulting. It provides a comprehensive introduction to use of the HP 3000 plus training in system management and operation. Key members of your staff may be trained at your site after system installation. All HP training and consultation services are available on a tuition basis and may be ordered from your HP Sales Representative. Training and consulting may be tailored to the specific needs of your installation in four different ways:

- Classroom courses taught at HP training centers, by professional instructors with a strong background in computer applications. These centers are equipped with HP 3000 Systems; courses are designed to provide a generous amount of practical, hands-on experience with the system.

- Self study courses employing modularized audio cassettes, student workbooks, and language reference manuals. The student gains practical "hands-on" experience by completing lab problems in each module, using an HP 3000 System at your site.
- Training and consulting conducted at your site by your HP Systems Engineer, after your system has been installed. Your S.E. has the knowledge and can obtain materials to teach certain training courses on your site.
- You can also purchase S.E. consulting in the form of on-site HP 3000 System Consulting. In all cases, S.E. consulting is for the purpose of conveying information, (i.e., conducting training, providing advice based on his experience and special knowledge, and directing you to specialized sources of information, etc.). The S.E. Consulting provided under this form of service never includes designing, coding, or converting the user's applications.
- All on-site training and consulting is subject to availability; the specific times during which the training and/or consulting are given at the installation site must be mutually agreed upon by you and your HP Sales Representative. Typically, the services will be used during a period not exceeding four months, commencing prior to the installation date of the system.
- HP 3000 Phone-In Consulting Service. When you purchase this service, you are provided with a special toll-free telephone number (where available) for obtaining answers to questions related to your use of the HP 3000 System.

HP 3000 TRAINING COURSES

Two training courses are fundamental to successful operation and usage of your system; the first, "HP 3000: A Comprehensive Introduction" is a five-day course available either at a Hewlett-Packard training center or on your site. This course is a prerequisite to all other 3000 training, and is a must for your System Manager and all programmers on your staff. The second fundamental course is "HP 3000: System Management and Operation." Your System Manager, and perhaps one alternate person, must attend this course (either on site or at a training center), to learn proper management and operation of your system.

Successful completion of these courses by your company's System Manager makes him eligible to take advantage of the Phone-In Consulting service and to submit "Software Problem Reports".

Listed below are HP 3000 Series II training courses and consulting services. Check with your HP Sales Representative for availability and course scheduling.

Installation Start-Up Package (on your site)

HP 3000: A Comprehensive Introduction

HP 3000: System Management and Operation

HP 3000: IMAGE (Data Base Management)

HP 3000: COBOL Self Study Course

HP 3000: BASIC Self Study Course

HP 3000: Advanced Usage Course

HP 3000: SPL (HP 3000 System Programming Language)

HP 3000: Data Entry Library

HP 3000: IMAGE Optimization

System/3 to HP 3000 Conversion, Costing Guide

HP 3000CX to Series II Upgrade

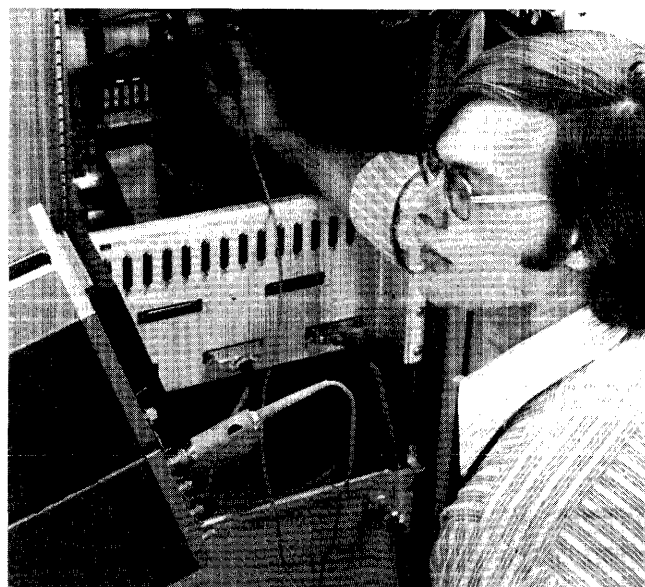
HP 3000 Installation Start-Up Consulting

HP 3000 System Consulting

Phone-In Consulting

HP 3000 SYSTEM MAINTENANCE PROGRAM

To help keep your installation running smoothly and to insure maximum up-time, Hewlett-Packard offers a basic maintenance service, with a series of options to tailor it to your particular needs. HP's System Maintenance Contract for the HP 3000 System, covers hardware and software maintenance, reference manual updates, and a subscription to the **Communicator**, a bimonthly software information service.



SYSTEM HARDWARE MAINTENANCE

The System Maintenance Contract covers the following hardware-related services:

- Six preventive maintenance visits are included per year. Additional visits may be necessary because of adverse operating environment or extreme heavy usage of particular peripherals (such as line printers); these visits are subject to an extra charge.
- Emergency repair of system failures. The System Maintenance Contract covers all costs of repairing or replacing any defective part of your system.

The HP Basic Monthly Maintenance Contract provides for typical 4-hour response times within a 100 mile radius of a major metropolitan area. Coverage is provided Monday through Friday, 8 a.m. to 5 p.m.

Your local HP Sales Representative can provide you with a quotation for extended coverage tailored to your specific needs. Extended service provides coverage for weekends and after 5 p.m. Extended coverage is not available in some locations. Check with your Sales Representative for details.

Hardware Maintenance does not cover:

Relocation or reconfiguration of the system

Damage caused by misuse of the system, air-conditioning failure or fire

Natural disaster

When you place an order for an HP 3000 System, a Customer Engineer in your local HP Office is assigned to your account. This Customer Engineer will be, in most cases, the same person who consults with you on site preparation, installs the system, and is personally held accountable for the maintenance, reliability and up-time of your system.

Your HP Customer Engineer has a personal, complete stock of spares for the most vulnerable parts of your system and a supply of the less critical items at regional service offices. The C.E. is also backed by the Customer Engineering group in your HP Sales Region and by the Product Support group at Hewlett-Packard, General Systems Division. This organization is geared to respond quickly with additional personnel and/or equipment if your C.E. has not been able to resolve a system problem within 4 hours of arrival on your site.

SYSTEM SOFTWARE MAINTENANCE

The System Maintenance Contract covers the following software-related services:

- Automatic software updates — When your HP Customer Engineer performs preventive maintenance, the latest updates of software purchased for your site, are installed on your system. This service also covers any unscheduled visits to install revised software to solve emergency software problems.

- Service on non-reproducible system failures — If the system should fail for no apparent reason, the HP 3000 central processor has a built-in memory dump facility that allows your system operator to dump the contents of memory to a magnetic tape, and restart the system — all in a matter of minutes. A system log is provided for your operator to enter this occurrence and note system user activity at the time of the failure.

- If this problem recurs frequently (i.e., more than once per week), your C.E. will collect the memory dump evidence, and with help from the HP 3000 Product Specialist, will attempt to isolate possible hardware or system software malfunctions. Your C.E. will send a report of the findings back to your company's System Manager, together with an action plan for resolving the problem.

REFERENCE MANUAL UPDATES

Under this service, which is part of the System Maintenance Contract, you receive one set of manual updates pertinent to the software purchased for your site. These updates are mailed directly to your company's System Manager.

If a software error is fixed or a product is enhanced, the appropriate reference manual is updated to reflect that change. Whenever an update to a manual is issued, your Company's System Manager will receive a set of replacement pages. These pages are provided in a convenient looseleaf format and can be easily inserted in the master set of system documentation. If changes are extensive, a reprinted copy of the manual will be sent.

If you require more than one set of manual updates, additional sets can be purchased through your local HP Sales Representative.

SOFTWARE INFORMATION SERVICE

This service is provided in the form of a publication called the **Communicator** which contains the following kinds of information:

- Lists of software problems fixed in the latest software updates
- Lists of known software problems, and "work-arounds" when available
- Training course schedules
- Lists of available software documentation, plus ordering information
- Announcement of the availability of new or updated manuals

- Application articles contributed by users, serving as a means of communications within the HP 3000 user community
- Feature articles of general interest

The **Communicator** is published by Hewlett-Packard, General Systems Division, six times per year, and distrib-

uted to customers by HP's Corporate Parts Center, which maintains the computer-based subscription service. The publication is printed simultaneously with the release of updated HP 3000 software and sent directly to your company's System Manager. Your System Maintenance Contract includes one subscription to the **Communicator**. Additional subscriptions for your site may be purchased through your HP Sales Representative.



PROGRAM SEGMENTING

*Jim Hooper
HP/Neely, Santa Clara*

If you are a newcomer to HP's RTE, or DOS systems, you may not yet be aware of the power that segmenting can bring to your programs. Even if you are an old hand at RTE, you may be missing a trick that can make segmenting easy.

Program segmenting is simply an orderly division of your program into smaller parts that are not core-resident at the same time. Program segmenting is done for you in the "virtual" memory machines, like the HP 3000. These machines are called virtual because the automatic program segmenting seemingly provides an endless amount of memory. On these machines you can write your program almost without regard to the actual memory available. If you use segmenting in RTE, the same virtually limitless resource of computer memory is available to you.

A "segment" in RTE is a program, which like a main program, has a "PROGRAM" statement in FORTRAN, an "HPAL" statement in ALGOL, or a "NAM" statement in Assembler. The program type is 5. It must be loaded by the relocating loader with a main program, type 3, background disc resident. When the segment is called, it is overlayed into memory behind the main program and main program subroutines. Unlike a swap, which requires 2 disc transfers, loading a segment requires only 1 transfer; it is twice as fast as scheduling a second program.

The load map produced by the loader shows that the main program and its subroutines are loaded first. Segments are loaded next. They are all relocated into memory at the same address, which is the next location after the last subroutine linked to the main program.

Using FORTRAN as an example, segmenting is implemented with a CALL EXEC(...). Parameters passed to the operating system in the EXEC call specify the segment load request code (8), segment name, and optionally up to 5 parameters. The operating system loads the segment and transfers control to the first executable line in the segment. Data can be transferred to the segment either through the optional parameters or through COMMON.

Two techniques to structure your programs using segmenting might be called Rippling and Return-to-Main techniques. The word Rippling is a coined expression to describe the technique of having one segment call another segment. Return-to-Main as the name implies, is the technique of having the main program call each segment, which then returns control to the main.

Rippling gives you the largest possible overlay area. As the sample program shows in Figure 1, the main program is very short. It might initialize variables, define COMMON, and call the monitoring segment. The monitoring segment evaluates the command or function and initiates the loading

of the next segment, which, in turn, may call another segment. When the command or function task is completed, the monitoring segment is reloaded. Each time the monitoring segment is loaded, execution begins at the first line.

The second technique, Return-to-Main, provides a structure similar to a subroutine, that makes program development or conversion easy. This technique is particularly effective for programs dependent on response time since its use might eliminate an extra segment load.

```

0001  FTN4,L
0002      PROGRAM MAIN
0003      COMMON IVALU
0004      DIMENSION ISEG1(3)
0005      DATA ISEG1/2HSE,2HG1,2H /
0006      CALL EXEC (8,ISEG1)
0007      END
0008  C
0009  C
0010  C
0011      PROGRAM SEG1,5
0012      COMMON IVALU
0013      DIMENSION ISEG2(3),ISEG3(3)
0014      DATA ISEG2/2HSE,2HG2,2H /
0015      WRITE(1,10)
0016  10  FORMAT(" SEGMENT 1. COMMAND?")
0017      READ (1,*) IVALU
0018  30  CALL EXEC(8,ISEG2)
0019  C    DUMMY CALL TO MAIN
0020      CALL MAIN
0021      END
0022  C
0023  C
0024  C
0025      PROGRAM SEG2,5
0026      COMMON IVALU
0027      DIMENSION ISEG3(3)
0028      DATA ISEG3/2HSE,2HG3,2H /
0029      WRITE(1,100)
0030  100  FORMAT(" IN SEGMENT 2.")
0031      CALL EXEC(8,ISEG3)
0032  C    DUMMY CALL TO MAIN
0033      CALL MAIN
0034      END
0035  C
0036  C
0037  C
0038      PROGRAM SEG3,5
0039      COMMON IVALU
0040      DIMENSION ISEG1(3)
0041      DATA ISEG1/2HSE,2HG1,2H /
0042      WRITE(1,1000) IVALU
0043  1000 FORMAT(" IN SEGMENT 3.  COMMAND = ",I5)
0044      IF (IVALU.EQ. 9) STOP
0045      CALL EXEC(8,ISEG1)
0046  C    DUMMY CALL TO MAIN
0047      CALL MAIN
0048      END
0049      ENDS
**** LIST END ****

```

Figure 1. Rippling Technique

In the Return-to-Main technique, the command or task monitor is part of the main program. The monitor loads the necessary segment, which returns control to the main at the next line after the segment load call. RTE does not provide a Return-to-Main capability as part of the segment load EXEC call, but it can be done easily through a simple programming trick.

There are a couple of programming tricks for segmenting that have already been used in RTE and DOS. One involves the FORTRAN "ASSIGN" statement and a short assembly language subroutine; the subroutine is a Contributed Library program, HP number 22431. The second trick, though similar, is more straightforward and is easier to use,

```

0001          ASMB,L
0002 00000      NAM LDSEG,7
0003          ENT LDSEG,RETRN
0004          EXT EXEC,,ENTR
0005 00000 000000 NAME BSS 1      STORAGE FOR PARAMETER ADDRESS
0006 00001 000000 LDSEG NOP      PRIMARY ENTRY POINT
0007 00002 016002X JSB ,ENTR     PICK UP PARAMETER ADDRESS
0008 00003 000000R DEF NAME
0009 00004 062000R LDA NAME      MOVE PARAMETER ADDRESS INTO THE
0010 00005 072011R STA ,NAME     SEGMENT LOAD EXEC CALL
0011 00006 016001X JSB EXEC     EXEC CALL TO LOAD SEGMENT
0012 00007 000012R DEF **3      CONTROL WILL NEVER REALLY
0013 00010 000014R DEF RCODE     RETURN TO **3, SEGMENT WILL
0014 00011 000000 ,NAME BSS 1    ACTUALLY DO A "CALL RETRN".
0015 00012 000000 RETRN NOP      RETURN TO THE MAIN PROGRAM BY
0016 00013 126001R JMP LDSEG,I   JUMPING THROUGH "LDSEG"
0017 00014 000010 RCODE DEC 8
0018*
0019*
0020*  USAGE INSTRUCTIONS:
0021*
0022*
0023*      IN MAIN PROGRAM      -  CALL LDSEG(ISEG)
0024*                               WHERE ISEG IS A 3 WORD
0025*                               ARRAY THAT CONTAINS
0026*                               THE SEGMENT NAME.
0027*      IN SEGMENT          -  CALL RETRN
0028*
0029*  NOTE THAT "LDSEG" CAN ONLY BE CALLED FROM MAIN.
0030*
0031          END
** NO ERRORS*

```

Figure 2. Load-Segment Subroutine

This trick is to save the return point to the main program in an assembly language subroutine. This subroutine then issues the segment load EXEC call. When the segment is finished, it jumps to the subroutine, and the subroutine returns to the main program. Because the main program calls our load-segment subroutine, the subroutine gets appended to the main. A second entry point in the subroutine prevents destruction of the main return address and allows the segment to link to the subroutine properly. Figure 2 is such a subroutine, which could be expanded to pick up optional parameters. Figure 3 shows the use of the subroutine in main and segment programs.

At load time, the loader imposes two requirements in order for the main and segments to be linked properly. If overlooked, these requirements will bring you frustration. Each segment must include a CALL statement that references the main program. This statement should not be executed; it is used for linking only. When called, LOADR must be told that it is to load a main program plus segments by setting the fourth LOADR parameter equal to 1.

Segmenting has always been a powerful tool. With the two techniques of Rippling and Return-to-Main, you can use segmenting even more effectively to solve your application problems.

```

0001  FTN4,L
0002      PROGRAM MAIN
0003      COMMON IVALU
0004      DIMENSION ISEG1(3),ISEG2(3)
0005      DATA ISEG1/2HSE,2HG1,2H /,ISEG2/2HSE,
0006          2HG2,2H
0006  5      WRITE(1,10)
0007  10      FORMAT(" MAIN    COMMAND?")
0008          READ(1,*)IVALU
0009          CALL LDSEG(ISEG1)
0010          CALL LDSEG(ISEG2)
0011          IF (IVALU ,NE, 9) GO TO 5
0012  20      STOP
0013          END
0014  C
0015  C
0016  C
0017      PROGRAM SEG1,5
0018      COMMON IVALU
0019      WRITE(1,50) IVALU
0020  50      FORMAT(" SEGMENT 1,  IVALU = ",I5)
0021          CALL RETRN
0022  C      DUMMY CALL TO MAIN
0023          CALL MAIN
0024          END
0025  C
0026  C
0027  C
0028      PROGRAM SEG2,5
0029      COMMON IVALU
0030      WRITE(1,80) IVALU
0031  80      FORMAT(" SEGMENT 2,  IVALU = ",I5)
0032          CALL RETRN
0033  C      DUMMY CALL TO MAIN
0034          CALL MAIN
0035          END
0036      ENDS
**** LIST END ****

```

Figure 3. Return-to-Main Technique

RTE-II/III CLASS TABLE STRUCTURE

Jim Bridges
HP Data Systems

One of the uses of system available memory (SAM) is to buffer class I/O requests as described in the **Communicator**, dated April 1, 1976. If the request is made to a physical device (LU not set to zero) then it is queued up on the device EQT. Upon completion of the request (LU set to zero implies immediate completion) the request is queued to caller through the class table.

It is not necessary to understand the structure of the class table to effectively use class I/O. However, situations may arise when buffers queued up in SAM may become "lost". This happens when a program terminates with one or more class I/O requests which have not been satisfied by a class "GET" in which the buffer is released, (caused by a programming error or through an abort operation).

If you know and understand the structure of the class table, you can write a program that "cleans up" SAM by obtaining the class number from the table and issuing the necessary "GET" to release the buffers.

System entry point \$CLAS contains the number of entries in the class table. The class table begins immediately after the location \$CLAS. The size of the table is determined by the answer to the generator prompt "# OF I/O CLASSES?" (Maximum is 255). Each entry is a 16-bit word which has one of the following states:

State 1: Class deallocated, available

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

State 2: Pointer to first entry in class queue

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	Address of buffer queued up on class number														

State 3: Class allocated, no one waiting. Number of pending requests counter may be 0 to 255.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	X	Security code					Number of pending requests							

State 4: Class allocated, someone waiting (suspended).
Number of pending requests counter may be 0 to 255.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	X	Security code					Number of pending requests							

Note: Bit 13 is a "DON'T CARE", i.e., not used by the system.

States 3 and 4 keep track of the number of uncompleted requests. If a table entry is in state 3 or 4, there are no buffers linked to that class. In state 4, someone has issued a "GET" before a class I/O request has completed. Perhaps a class request has not even been made. Class requests which have been made but are not complete are queued on the EQT of the device which was referenced in the call.

All class requests use SAM. Uncompleted requests are stacked on the EQT. When the request is completed, the system removes the linkage to the EQT and establishes the link to the class table. The completed request then waits for a class "GET" to pick up the information and release (optionally keep) the buffer.

Completed requests are stacked on a first in — first out (FIFO) basis. The class table entry is in state 2 if there are any requests waiting for "GET". The class word (looks like state 3 or 4) is placed on the tail of the stack. Thus a class table entry in state 2 may be stacking a pending queue as well as a queue of completed buffers.

The class number which the system allocates to a program contains two elements:

1. Bits 0—7 contain the index into the class table, i.e., the first position in the table has an index value of 1. Class numbers are arbitrarily issued from the high end of the table.
2. Bits 8—12 contain the program's index into the keyword block, module 32. That is, the security code is formed by masking the low 5 bits of the program's ID segment number.

A program using the class number may add bits 13, 14 and 15 (as noted in the manual) but these bits are not used to identify the caller. The system uses the class table index in bits 0—7 of the caller's class number to find the table entry. If the security code does not match the caller, he is aborted with IO00 error. Actions taken by the system when handling a class I/O request depend upon the current state of the class table entry.

If a GET request is made and the entry is in:

- State 1 — The system aborts the program with IO00 error. No class number has been given.
- State 2 — The system returns the data from the class buffer.
- State 3 — The system sets the "someone waiting" bit (14). Suspend that program.
- State 4 — The system aborts the program with IO00 error. Only one program may be suspended per class.

If a class I/O request (READ, WRITE or WRITE/READ) is made and the entry is in:

- State 1 — State 3 is set up. Security code is low 5 bits of program's index into keyword block. Counter is set to 1.

State 2 — Counter at end of completed queue is incremented by 1.

State 3 — The counter is incremented by 1.

State 4 — The counter is incremented by 1.

On completion of a class I/O request (READ, WRITE or WRITE/READ), if the table entry is in:

State 1 — This is illegal and, should never happen. Buffer is returned and the completion is ignored.

State 2 — The new data is added at the end of the list (FIFO) and the counter is decremented by 1.

State 3 — The new data is added at the end of the list (FIFO) and the counter is decremented by 1.

State 4 — The waiting program is scheduled and the counter is decremented by 1. The "someone waiting" bit (14) is cleared.

A program suspended for a "GET" has its class number (bits 0–12) in word 2 of its ID segment. Only one program may be suspended on a class: more than one would result in ambiguous situations since only one bit is used to indicate someone waiting.

With the information provided here, a program can be written to examine the class table. The class number of any entry can be created by merging the table index with bits 8–12 of the entry. The completed queue may be "unstacked" (released) by using the number to issue class "GETS" until the system deallocates the class number (i.e., reduces the table entry to zero). Possible owners of the class number can be identified by the security code. If there are less than 32 ID segments, the owner can be uniquely identified. If someone is waiting on a class, the person waiting can be found by looking for the class number in word 2 of the ID segments of programs in the general wait list (status = 3).

THE WHZAT PROGRAM

Sandy Martensen
HP General Systems

The following article on the WHZAT program is taken from RTE III: A Guide for New Users (HP Part No. 92060-90012). It appears in this issue of the **Communicator** in response to numerous requests for more information on WHZAT.

WHZAT describes the current RTE-III system environment. It operates in one of two modes which you select when you type the RUN command. One mode displays all scheduled and suspended programs and their status and the other displays in numeric sequence all partitions with their status.

The parameters for running WHZAT are:

RUN,WHZAT *lu,option

Where:

lu = the logical unit number of the device on which you want the information displayed, the default value is LU 1, the system console.

option = indicates what you want displayed.
0 (the default value) will cause the status of scheduled and suspended programs to be displayed.
A non-zero value will cause the status of all partitions being used to be displayed.

Both parameters are optional. If you do not supply them, the status of scheduled and suspended programs is displayed on the system console.

In either mode, the information is preceded by a heading which includes the current system time and column headings. Figure 1 contains a sample printout of the program status mode and Figure 2 shows a sample of the partition status mode.

You may find yourself in the following situations:

1. Your program is in the WAIT state because an input/output device is down and RTE cannot initiate an I/O operation requested by your program.

WHZAT lists the program in state # with EQ DOWN. Examine the list of DOWN EQT's at the bottom of the program status report to see which devices are down. After you make the device ready, use the ***UP,eqt** command to notify the system that the device is now operating properly.

2. Your program is in the WAIT state because a logical unit it wants to use is locked to another program, LULK ***lu***, LKPRG = program name.

You can look at the status of the program which is using the device and decide to wait until the program has finished with it or give up for now and terminate your program with ***OFF, program name,1**. If the other program's status indicates that it is having problems with the device, you can notify the system manager or the person who is running the program so they can decide what to do. They may decide to terminate that program and then yours will continue executing.

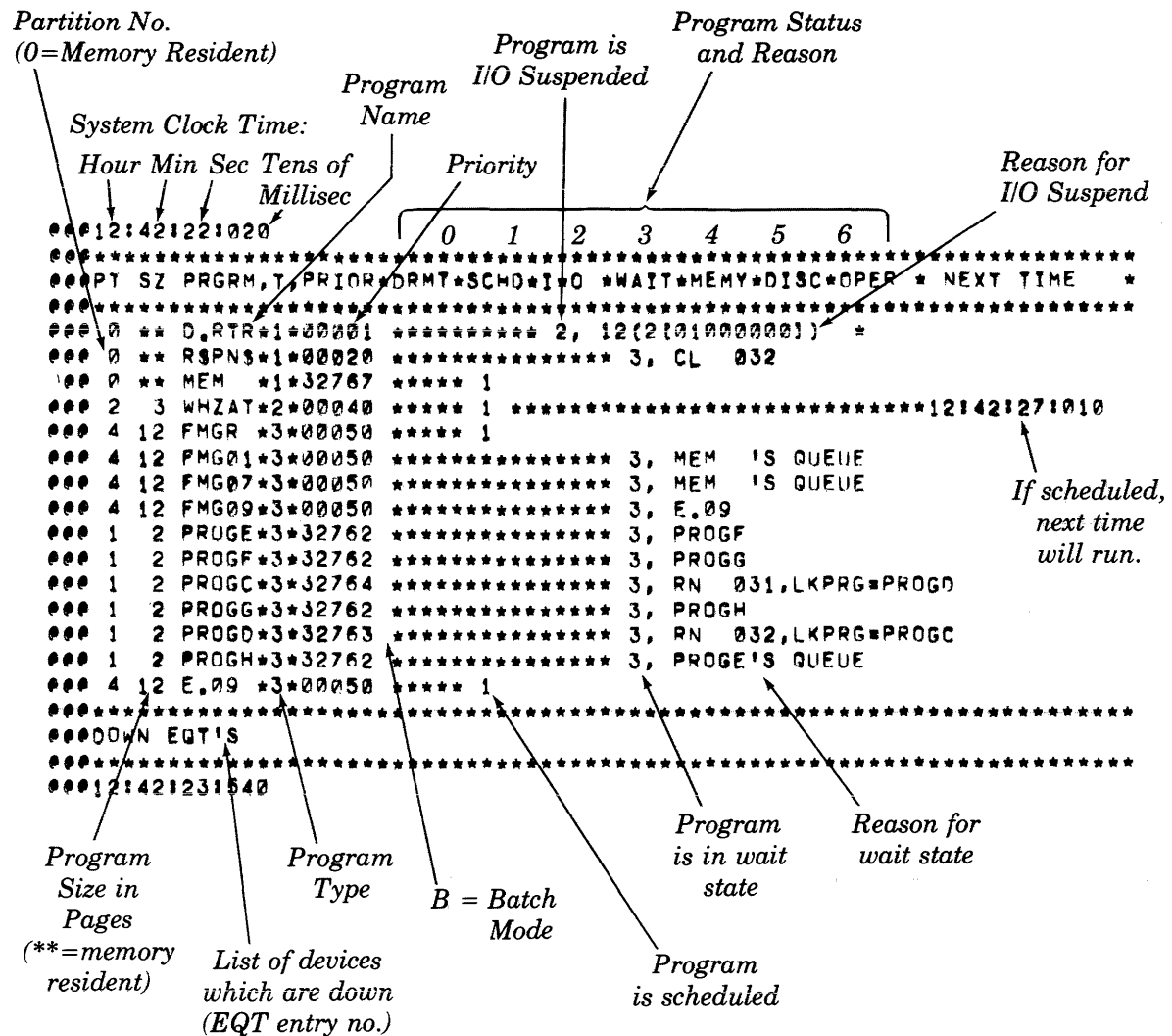


Figure 1. Program Status Mode (*RUN,WHZAT)

System Clock Time:

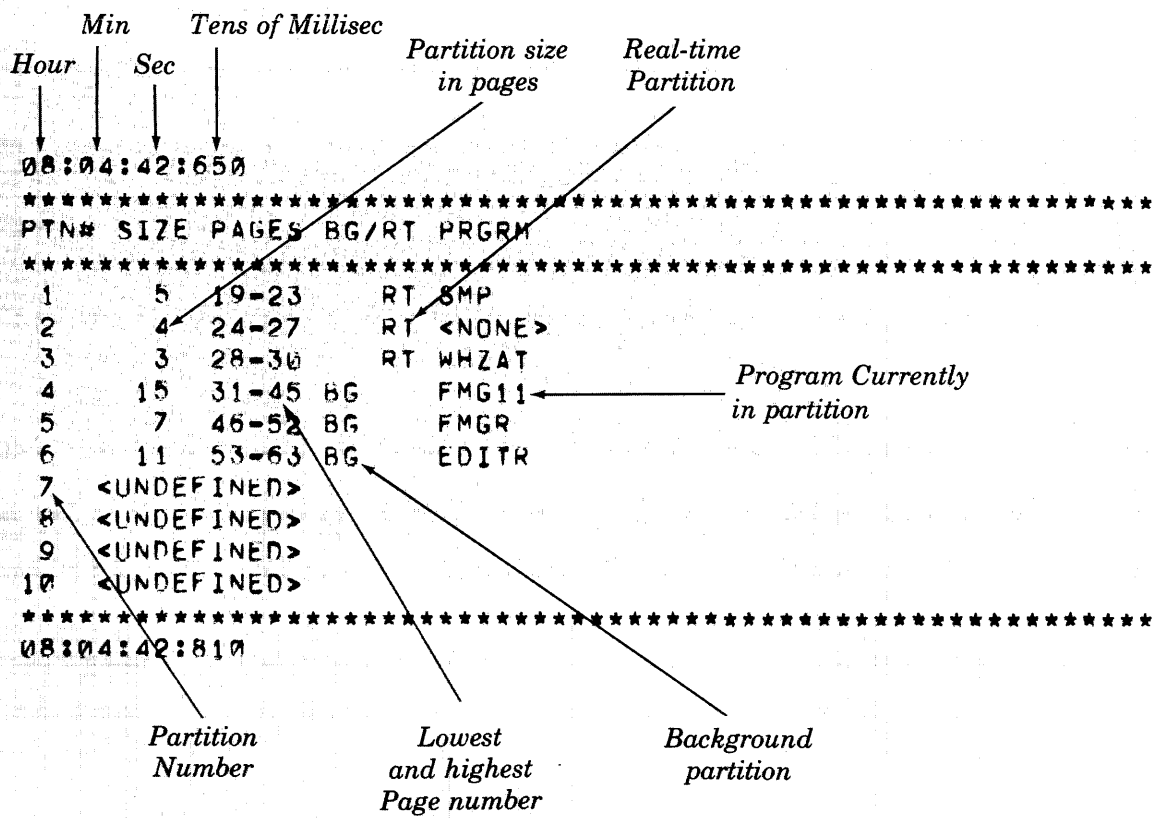


Figure 2. Partition Status Mode (*RUN,WHZAT,,1)

3. Your program is in the WAIT state and waiting for another program to complete or waiting to begin running another program which is busy.

Check the status of the other programs and decide whether to wait or terminate your program and try again later. You must be careful when you terminate programs with the 1 parameter. If someone is using FMGR to pack the disc and you terminate FMGR with a 1, you can destroy the disc content. You should always determine that no one else is using FMGR for this purpose or for running a batch operation before terminating it with 1.

4. Your program is I/O suspended because a device is down.

Check the DOWN EQT's list and fix the device. Then use the *UP_{eqt} command to notify RTE that the device is ready. (This situation is similar to the WAIT state EQ DOWN but occurs when a device goes down while you are using it rather than before I/O is initiated.)

5. Your program is I/O suspended because a device is busy.

You must either wait for the device to be free or terminate your program and try later.

6. Did you accidentally omit a parameter when typing a command so that the default logical unit number is not the device you really want to use? For example:

*RU,LOADR The LOADR will try to read from the standard input device (LU 5) instead of the LG area since you did not specify the 99 parameter. This may cause your program to be I/O suspended.

The partition status mode provides a dynamic map of the activity in each partition and the partition size.

"NO ABORT" RETURN FROM FORTRAN SUBROUTINES

Jim Bridges
HP Data Systems

Several FORTRAN subroutines on HP RTE systems have a "no abort" option, which you can select by setting bit 15 on the request code. An example is the resource management routine RNRQ.

The following example illustrates difficulties that can be encountered when trying to recover the A and B registers via the "standard" procedure.

```

.....
DIMENSION IREG (2)
EQUIVALENCE (IREG,REG,IA),(IREG(2),IB)
.....
ICODE = 140044B
REG = RNRQ (ICODE,IRN,ISTAT)
C THE FOLLOWING "GO TO" WILL ALWAYS
EXECUTE
C ALTHOUGH IT IS SUPPOSED TO BE AN ERROR
RETURN
GO TO 100
C THE FOLLOWING CODE WILL NEVER BE
EXECUTED
200 .....
.....

```

In this example, the routine RNRQ is called as if it were a real function (rather than a subroutine). The compiler generates a code sequence appropriate for all real functions as follows:

```

JSB      RNRQ
DEF      *+4
DEF      ICODE
DEF      IRN
DEF      ISTAT
JSB      .DST
DEF      REG
JMP      <GO TO 100>
.....

```

Note that we have a JSB to store the A and B registers exactly where RNRQ expects to find an error return. This results in the program always going to the error routine and saving A and B registers only if there is an error!

This is not a bug, but results from an inappropriate use of a function call.

One way around this problem is to define an assembly language subroutine which would return the A and B registers as follows:

```
CALL ABREG (IA,IB)
```

where IA and IB hold the A and B registers upon return from the subroutine.

KNOW YOUR RTE, PART 2

This is the second of a series of articles in the **Communicator** dealing with the inner works of HP's RTE systems. Hopefully, this information on how RTE works will enable you to use your system more efficiently and get more results with less effort.

In these articles we go into some detail on how the system works; therefore, you should have already read and become familiar with the material in the RTE reference manual applicable to your system. We will try to present this information in a way that eliminates forward references, but this will not always be possible. Each article will build on previous articles so that we can get into the system in some depth.

In part one of this series printed in the **Communicator** dated April 1, 1976, we described the list processor. In looking over the article, a user pointed out that the sequence:

```

*SS,X      (operator suspend X)
*GO,X      (now let X continue)

```

does not always put the program back to the same state. For example, a program in the general wait state waiting for anything but a 'son' to complete is in state 3, substate 4, which the 'SS' command maps to state 6, substate 0. The 'GO' command then maps this state to 1, substate 0, and not to state 3, substate 0 (see Figure 1). The user tried it on his system and found that the program really ended up in state 3, substate 0. He then informed us that our table was in error and that the system must in fact have more to work with than the state, substate information since he could find no way to encode the required information in the available bits.

What happened?

The table is right. The user's observations are also right. What is missing is information on a convention the system uses for most requests which end up suspending a program. *The convention is to leave the point of suspension at the JSB of the user's request. This means that when the program is next run it will remake the same request.*

In the example just described the 'GO' request did in fact move the program to state 1, substate 0. The program was then dispatched, remade the same request that put it into state 3, substate 0 and, since nothing else had changed, was put right back in state 3, substate 0. All of this happened so fast that the user did not see the program in state 1. The system does have more to work with than the state, substate. It has the whole program and its point of suspension.

In the rest of this article we will discuss system startup using a "bootstrap" loader.

SYSTEM START UP

"Bootstrap" is a technique for loading the first few instructions of a routine into storage, then using these instructions to bring in the rest of the routine.

The system gets loaded into memory by a boot extension which resides on logical track 0 of the system disc. In fixed head systems this extension is in two parts each 64 words long, while in moving head systems it is 128 words long and in one part. When the system is actually being loaded into memory this boot extension resides just below the BBL or BBDL. (BBL is Basic Binary Loader; BBDL is Basic Binary Disc Loader). All BBL, BBDL's are constrained to reside in the last 64 memory locations of the CPU's actual (i.e., unmapped) address space. The boot code is from X7500g through X7677g for the moving head disc systems, and X7600g through X7677g for the fixed head disc systems. X depends on the memory size of the system (see Figure 3). The boot extension itself is brought into memory by either a BBDL or by a paper tape.

FIGURE 1.

THE ANNOTATED SYSTEM

J ■ JMP ENT
 S ■ SUB ENT
 V ■ VARIABLE
 A ■ ADDRESS
 X ■ DANGEROUS
 ? ■ UNKNOWN?

DISPA(0000)02000 03165 92001-16012 REV.A 740801
 J \$RENT 02164 RETURN FROM \$LIBX/\$LIBR FOR PROGRAM DISPATCH
 S \$BRED 03057 SUBROUTINE TO LOAD A SEGMENT
 V \$ZZZZ 03115 TERMINATION STACK HEAD
 J \$XEQ 02036 WHERE TO GO WHEN YOU HAVE NOTHING TO DO.
 DISPM ADDITIONAL ENTRY POINTS
 A \$MRMP ADDRESS MEMORY RESIDENT MAP. SET UP BY
 GENERATOR.
 V \$ENDS PAGES OCCUPIED BY SYSTEM AND RESIDENT
 LIBRARY. SET UP BY GENERATOR. DOES NOT CHANGE.
 A \$MATA ADDRESS FIRST ENTRY MEMORY ALLOCATION TABLE.
 SET BY GENERATOR.
 A \$MPFT ADDRESS MEMORY PROTECT FENCE TABLE. SET
 BY GENERATOR.
 V \$BGFR LIST HEADER FOR FREE BG PARTITIONS. SET
 UP INITIALLY BY GENERATOR.
 V \$RTFR LIST HEADER FOR FREE RT PARTITIONS. SET
 UP INITIALLY BY GENERATOR.
 S \$ALDM SUBROUTINE TO REMOVE A PARTITION FROM
 ALLOCATED LIST AND LINK INTO DURMANT LIST.
 S \$DMAL SUBROUTINE TO REMOVE A PARTITION FROM DORMANT
 LIST AND LINK INTO ALLOCATED LIST.
 S \$SMAP SUBROUTINE TO LOAD USER MAP.
 S \$SPRCN SUBROUTINE TO RELINK PROGRAM IN ALLOCATED
 LIST BY NEW PRIORITY (AFTER PR CHANGES).
 A \$EMRP LAST WORD ADDRESS MEMORY RESIDENT PROGRAM
 AREA. SET BY GENERATOR.
 S \$XDMP SUBROUTINE FOR NON-PRIVELEDGED DRIVER TO
 CALL TO SET UP USER MAP (2313 USES).
 RTIME(0000)03166 03737 92001-16012 REV.B 741022
 S \$TADD 03637 ADD PROGRAM TO TIME LIST
 J \$CLCK 03166 TEN MS TICK ENTRY POINT
 S \$TREM 03661 REMOVE PROGRAM FROM TIME LIST
 V \$TIME 03370 SYSTEM TIME (10'S MS DAY (2 WORDS), DAY/YEAR (1 WORD))
 S \$TIMV 03436 CONVERT TIME TO 6 WORD ARRAY
 S \$ETIM 03604 CONVERT 4 WORD TIME TO 2 WORD TIME
 J \$TIMR 03532 EXEC 12 COMPLETION PROCESSOR
 J \$ONTM 03503 ON REQUEST COMPLETION PROCESSOR
 J \$TMRQ 03707 COMPLETES THE TM REQUEST
 S \$SCLK 03407 START SYSTEM CLOCK SUB. (USED BY PFAIL)
 V \$BATM 03365 BATCH TIMER (2 WORD 10'S MS)
 SASCH(0000)03740 04017 92001-16012 REV.A 740801
 V \$OPER 03770 OP CODE ERROR
 V \$ERIN 04010 INPUT ERROR
 V \$NOPG 04000 NO SUCH PROGRAM
 V \$ILST 03740 ILLEGAL STATUS
 V \$NOLG 03751 NO LOAD AND GO SPACE
 V \$LGBS 03761 LOAD AND GO IN USE
 RTIOC(0000)04100 07723 92002-16022 REV.B 741022
 S \$CIC 04100 EVERY NON PRIVILEGED INTERRUPT COMES HERE
 S \$XSIO 05744 SYSTEM I/O ROUTINE (SYS TTY, DISC, SPOOL)

S \$SYMG 07040 SYSTEM MESSAGE TO TTY
 J \$IORQ 04267 EXEC CALL FOR I/O (1,2,3,13,17,18,19,20)
 J \$IQUP 07025 UP MESSAGE PROCESSOR ENTRY
 J \$IODN 06775 DN MESSAGE PROCESSOR ENTRY
 S \$ETEQ 07151 SETS UP EQT1 - 15
 J \$IRT 04216 RESTORE STATE AND RETURN TO PROGRAM
 J \$XCIC 04114 ALTERNATE POST CLF ENTRY TO SCIC
 J \$DEVT 06717 DEVICE TIMED OUT ENTRY (FORM RTIME)
 J \$GTIO 05264 EXEC 21 (GET) ENTRY POINT
 J \$UPIQ 07027 DRIVER UP ENTRY POINT
 S \$CVEQ 07127 CONVERT EQT # TO ADDRESS AND SET UP ON BP.
 J \$YCIC 04115 ALTERNATE POST CLF AND LIA 4 ENTRY TO SCIC
 V \$BLLO 00112 LOWER BUFFER LIMIT
 V \$BLUP 00113 UPPER BUFFER LIMIT
 J \$IOCL 07210 I/O CLEAR ENTRY (FROM OF)
 J \$LUPR 07325 LU REQUEST
 J \$EQST 07473 EQ REQUEST
 J \$CHTO 07561 TO REQUEST
 RTIOC ADDITIONAL ENTRY POINTS FOR RTE III
 X \$DVM THIS ROUTINE SETS UP AND ENABLES THE CORRECT
 MAP NECESSARY TO ENTER A DRIVER. THIS ROUTINE
 SHOULD NOT BE CALLED BY ANYONE EXTERNAL TO RTIOC.
 THE ONE EXCEPTION IS THE SPOOL DRIVER.

 X \$RSM THIS ROUTINE RESTORES THE USER MAP, IF IT
 WAS CHANGED, TO ITS STATUS PRIOR TO ENTERING A
 DRIVER. THIS ROUTINE SHOULD NOT BE CALLED BY ANYONE
 EXTERNAL TO RTIOC. THE ONE EXCEPTION IS THE SPOOL
 DRIVER.

 V \$MEU MEU STATUS AT INTERRUPT. IT IS USED BY
 THE MESSAGE PROCESSOR IN SCHED.
 \$ALC (0000)07753 10160 92001-16012 REV.A 740801
 S \$ALC 07753 ALLOCATE MEMORY
 S \$RTN 10044 DEALLOCATE MEMORY

 EXEC (0000)10164 11720 92001-16012 REV.B 741022
 ? EXEC 10164 GUESS WHAT
 S \$ERMG 11554 SEND ERROR MESSAGE AND ABORT
 J \$RQST 10166 MP VIOLATION PROCESSOR
 J \$OTRL 11376 COMPLETE THE RT COMMAND
 S \$LIBR 10374 PRIV/RENT SUBROUTINE ENTRY
 S \$LIBX 11055 PRIV/RENT SUBROUTINE EXIT
 S \$DREQ 11430 ALLOCATE N TRACKS
 S \$DREL 11514 RELEASE N TRACKS
 S \$SDRL 11403 RELEASE ALL TRACKS OWNED BY THIS PROGRAM
 S \$SDSK 11532 SCHEDULE ALL DISC SUSPENDED PROGRAMS
 J \$ERAB 11371 ADD "00" TO B, CALL \$ERMG, GO TO \$XEQ
 V \$PVCN 10506 PRIVILEGED COUNTER
 S \$REIO 10642 CHECK IF I/O BUFFER IN TDB AND MOVE IF SO
 S \$CREL 11272 CONDITIONAL TRACK RELEASE (LG / GLOBAL)
 S \$RSRE 10711 RESTORE ALL TDB'S FOR CURRENT PROGRAM
 S \$ABRE 10776 RETURN ALL RENT MEMORY (CALLED ON TERMINATION)
 V \$PWR5 10247 PFAIL MUST: LIA 5 / STA \$PWR5 WHEN GOING DOWN
 *XECM ADDITIONAL ENTRY POINTS FOR RTE III
 V \$MVBF TDB MOVED TO SYSTEM AVAILABLE MEMORY FLAG.
 IT IS USED BY RTIOC TO DETERMINE WHICH MAP IS
 NECESSARY FOR AN I/O CALL FROM A PROGRAM WITH
 RE-ENTRANT BIT SET.

 A \$SGAP START ADDRESS OF SUBSYSTEM GLOBAL AREA.
 IT IS SET UP BY THE GENERATOR AND IS USED FOR

THE UPPER BOUNDS TEST FOR A RESIDENT LIBRARY CALL.

```

*TRRN(0000)11726 12071 92001-16012 REV.A 740801
S  STRRN  11726  RELEASE/ RETURN ALL LU LOCKS/ RN'S
S  SCGRN  11772  DRIVER ENTRY TO CLEAR GLOBAL RN
S  SULLU  12014  UNLOCK ALL LU'S

SCHED(0000)12135 15177 92002-16012 REV.B 741027
S  $LIST  12177  LIST PROCESSOR
S? $MESS  12476  SYSTEM COMMAND PROCESSOR
S  $CVT3  14273  CONVERTS A NUMBER TO ASCII (6 DIGITS)
S  $CVT1  14337  CONVERTS A NUMBER TO ASCII (2 DIGITS)
S  $ABRT  14426  ABORT SUBROUTINE
J  $TYPE  14343  SOMEBODY HIT A KEY ON SYS. TTY
J  $MPT1  14470  EXEC 6 COMPLETION (TERMINATION)
J  $MPT2  14626  EXEC 7 COMPLETION (SUSPEND)
J  $MPT3  14637  EXEC 8 COMPLETION (SEGMENT LOAD)
J  $MPT4  14727  EXEC 9,23 COMPLETION (SCHEDULE WITH WAIT)
J  $MPT5  14747  EXEC 10,24 COMPLETION (SCHEDULE WITHOUT WAIT)
J  $MPT6  14766  EXEC 11 COMPLETION (GET SYSTEM TIME)
S  $PARS  12610  SYSTEM COMMAND PARSE ROUTINE
J  $STRT  13430  START SYSTEM HERE
S  $SCD3  14705  RESCHEDULE ALL PROGRAMS SUSPEND BECAUSE ...
J  $INER  14005  SEND INPUT ERROR MESSAGE
J  $MPT7  15020  EXEC 12 COMPLETION (PUT PROGRAM IN TIME LIST)
V  $ASTM  12541  TM ENTRY IN COMMAND TABLE
J  $MPT8  15144  EXEC 22 COMPLETION (SET/CLEAR CORE LOCK/MEMORY FLAGS)
S  $IDNO  15005  GET THIS PROGRAMS ID SEGMENT NUMBER
V  $WORK  12145  LAST PROGRAM PROCESSED BY $LIST (ID ADDRESS)
S  $WATR  14616  SCHEDULE PROGRAMS QUEUED TO SCHEDULE THIS ONE
V  $IDSM  14057  SMP'S ID SEGMENT ADDRESS IF ANY (USED BY RTIME)
SCHEDM  ADDITIONAL ENTRY POINTS FOR RTE III
V  $MPSA  SYSTEM AVAILABLE MEMORY WORD, BITS 0-9 HAVE
          THE STARTING PAGE OF SAM, AND BITS 10-15 HAVE THE
          NUMBER OF PAGES. IT IS SET UP BY THE GENERATOR AND
          IS USED DURING SYSTEM INITIALIZATION TO BUILD THE
          SYSTEM MAP.

J  $MSEX  THE RETURN POINT FOR ALL COMMANDS GOING THROUGH
          THE MESSAGE PROCESSOR. IT IS IMPERATIVE ALL RETURNS
          GO THROUGH THIS POINT BECAUSE IT RESTORES THE STATUS
          OF THE MEU AT INTERRUPT.

DVP43(0000)15263 15610 92001-16004 REV.B 741028
S  $POWR  15263  POWER FAIL ENTRY POINT
S  IP43   15556  I. ENTRY FOR POWER FAIL (GET TIME OF FAILURE)
S  CP43   15460  C. ENTRY FOR PFAIL (TIME OUT ENTRY FOR EQT PROCESSING)

```

BBDL Operation (Moving Head Disc)

All Basic Binary Disc Loaders on moving head disc systems load sector 0, track 0, of the specified disc surface into memory location 2011g and transfer control to it at the address pointed to by its relative location 44g. That is, after the sector is loaded into memory, location 2055g (2011g + 44g) points to the actual starting address of the bootstrap. Up to this point all HP moving head BBDL's are identical. The following is unique to RTE systems.

The disc's extension, when it gets control, clears the I/O system and stops at halt 77. This halt allows the user to protect the loader (if required) and also allows the 'slow boot' described later. When you push 'RUN' the boot moves itself up to its actual location in memory just below the BBDL and then transfers control to itself at this new location.

FIGURE 2.

SLIST STATE TRANSITION TABLE:

THE FOLLOWING TABLE DETAILS THE STATE TRANSITIONS EFFECTED BY SLIST. THE MAJOR STATES ARE 0 THRU 6 (DORMANT THRU OP-SUSP) AND THE STATE MODIFIERS ARE THE ADDITIONAL BITS SET FROM TIME TO TIME IN THE STATUS WORD. THE BITS WHICH AFFECT OR ARE MODIFIED BY SLIST ARE (SEE ABOVE DESCRIPTION):

BIT	WEIGHT	POSITION	
O	10	9	OPERATOR SUSPEND
W	4	12	WAIT STATE
R	2	7	RESOURCE SAVE
D	1	6	TO BE SET DORMANT AT NEXT SCHEDULE

THESE BITS ARE COMBINED TO FORM 16 SUBSTATES AS PER THE STATE TABLE. THE ENTRIES IN EACH SQUARE OF THE TABLE DEFINE THE NEXT STATE AS FOLLOWS:

THE FIRST DIGIT IS THE REQUESTED MAJOR TRANSITION (FROM THE SLIST CALL).
THE SECOND TWO NUMBERS (SEPERATED BY A ".") DEFINE THE NEXT MAJOR STATE, SUBSTATE. THUS 62.10 INDICATES A OP-SUSPEND REQUEST (6) CAUSES A MOVEMENT TO I/O SUSPEND (2) SUBSTATE 10 (THE O BIT IS SET).

A "*" AS THE DESTINATION INDICATES THE CURRENT STATE/SUB-STATE I.E. NO CHANGE.

ILLEGAL OR UNEXPECTED STATES ARE MARKED WITH "X"

ONLY EXPECTED CALLS ARE PLOTTED.

IN GENERAL CODE EXTERNAL TO SLIST MOVES PROGRAMS FROM SUB-STATE TO SUB-STATE WHILE ONLY SLIST CAN MOVE A PROGRAM FROM ONE MAJOR STATE TO ANOTHER.

Paper Tape Bootstrap

The paper tape bootstrap resides in memory locations 3 through 100g. When 'RUN', it loads the boot extension directly to its proper location just below the BBL area and transfers control to it.

BBDL Operation (Fixed Head Disc)

The fixed head BBDL loads sector 0, track 0, to location 0 through 100g on the CPU base page, and transfers control to it at location 77g. This code then does a halt 77 to allow the user to protect the BBDL and set the switch register. When RUN, it checks the switch register (0 for RTE) and loads the appropriate boot extension to memory just below the BBDL, and transfers control to it.

Boot Extension Operation

After the boot extension is in memory at the proper location (just below the BBDL or BBL) it gets control and loads the system into memory. The system start address was put in location 3 by the generator (along with a JMP,3,I in location 2). After a successful load, the boot extension transfers control to the system by executing a JMP 3,I.

The Slow Boot

Occasionally you may want to bring the system into memory and modify it before it starts. Now that you know how the system is booted we can define a SLOW BOOT process that keeps control so that modifications can be made to the system and/or the boot routines before they are run. The basic steps are as follows:

- A. LOAD THE 'BOOT' (i.e., DO A NORMAL BOOTUP TO THE FIRST HALT). THE 'BOOT' CAN NOW BE MODIFIED.
- B. PLACE A HALT INSTRUCTION AT THE LOCATION THAT JUMPS TO THE 'BOOT-EXTENSION!'
- C. RUN THE 'BOOT'
- D. THE 'BOOT' SHOULD STOP AT THE HALT YOU INSERTED. THE 'BOOT-EXTENSION' IS NOW IN MEMORY AND CAN BE MODIFIED.
- E. PLACE A SECOND HALT INSTRUCTION, THIS TIME IN THE 'BOOT-EXTENSION' AT THE LOCATION THAT JUMPS TO THE SYSTEM.

FIGURE 2 (CONT.)

SYSTEM STATE TABLE

SYSTEM STATE TABLE

MAJOR STATE 0

1

2

3

4

5

6

SUB-STATES

0	11.0	00.0	02.1	00.0	00.0	00.0	00.0
		22.0	11.0	11.0	11.0	11.0	11.0
		33.0	62.10	66.0	66.0	66.0	
		44.0					
		55.0					
		66.0					
1 D	X	X	02.1	X	X	X	X
			10.0				
			62.11				
2 R	11.0	00.2	02.3	00.2	00.2	00.2	06.3
	66.3						
3 RD	X	X	0*	X	X	X	0*
			10.2				10.2
4 W	00.0	33.4	00.0	00.0	00.0	00.0	00.0
				1*			13.4
				66.4			
5 WD	X	X	X	X	X	X	X
6 WR	0*	X	X	00.6	X	X	06.7
	13.4						
	66.7						
7 WRD	X	X	X	X	X	X	0*
							10.6
10 O	X	X	02.11	X	X	X	X
			16.0				
			6*				
11 OD	X	X	0*	X	X	X	X
			10.0				
			6*				
12 OR	X	X	02.13	X	X	X	X
13 ORD	X	X	0*	X	X	X	X
			16.3				
			6*				
14 OW	X	X	X	X	X	X	X
15 OWD	X	X	X	X	X	X	X
16 OWR	X	X	X	X	X	X	X
17 OWRD	X	X	X	X	X	X	X

SYSTEM DISC TYPE (by driver number)	BOOT TYPE	STEP			
		B (boot address)	C Run From	E boot-extension JMP-sys-loc	F boot-extension start address
DVR 30	BBDL	36 ₈	P- at HLT	X7605	X7600
DVR 31 (Moving Head Disc)	Paper tape	62 ₈	100 ₈	X7605	X7500
	ROM/BBDL	N/A**	N/A**	2016 ₈	P- at HLT
DVR 32 (Moving Head Disc)	Paper tape	44 ₈	100 ₈	X7505	X7500
	ROM/BBDL	N/A**	N/A**	2016 ₈	P- at HLT

**Steps B and C are not applicable here because Step A brings in the boot-extension directly.

Figure 3

Where: X depends on memory size as follows:

Memory Size	X
RTE III OR 32K	7
24K	5
16K	3

- F. RUN FROM THE START OF THE 'BOOT-EXTENSION'.
- G. THE 'BOOT-EXTENSION' SHOULD STOP AT THE HALT YOU INSERTED.
- H. THE SYSTEM IS NOW IN CORE AND CAN BE MODIFIED.
- I. START THE SYSTEM AT LOCATION 2.

Refer to Figure 3 for the addresses used in Steps B, C, E, and F of the Slow Boot procedure.

The System Starts

The address put in location 3 by the generator is the address of \$STRT which is currently located in the SCHED/SCHEDM module (refer to the Annotated System, Figure 2). In order to bring up the system, certain tables need to be set up and certain things need to be initialized. We will trace through the things to be done, not necessarily indicating who does them, but we will try to cover them in the same order that they take place at start up.

The RTE-II generator sets up some six different areas in memory which the system collects into the system buffer memory pool. (See Figure 6-5 in your RTE-II manual). The RTE-III generator also sets up a system buffer pool but it only sets up one area. The generators pass the addresses and the number of words (may be zero) in each area in base page locations labeled EQT 1 through EQT 12 (refer to Appendix A of your RTE-II/III manual). The memory manager (\$ALC) is coded with a pre-initialized empty memory pool. One of the first things the system has to do is to gather up the memory referred to in EQT 1 through EQT 12 and return it to the memory pool by making standard return calls to the memory manager. Prior to doing this, however, the RTE-III system must set up and enable the system map so that the system memory pool is addressable. Whenever memory is returned the memory manager updates its 'max memory now' which indicates the largest contiguous current block of memory.

After the system has returned all the memory set aside by the generator, this 'max memory now' flag represents the 'max memory ever'. The system saves this information for future reference in memory allocation. (We will cover the memory manager in detail in a future column.)

The RTE-III system at this time initializes the user map registers, by setting up the core resident user's map, and transfers the system map to both the DCPC MAPS. The RTE-III system also sets up pointers to its memory allocation table at this time. The rest of the system initialization is the same for both RTE-II and RTE-III. Next the 'swap delay' from the generator is extracted from the base page word 'SWAP', made negative and saved for the dispatcher. The system computes the minimum track size of the two possible system discs (LU2/3) and saves this track size so it will know how many tracks to allocate to swap any given program.

At this time the system start up code attempts to schedule FMGR. If the schedule is successful (i.e., if FMGR exists) FMGR's priority is set to 0 and its real priority is set in its temporary word number two (in its ID segment). The system also sets the track assignment table length to minus 1 (i.e., 1 track) and sets the true track assignment table length in temporary word 1 of FMGR's ID segment.

These things are done to prevent any other program from destroying files while FMGR sets up the file system. FMGR will reset its priority and the track assignment table length when it feels it's safe. The system now calculates the ID segment addresses for FMGR, D.RTR, EDIT, and SMP and puts these aside for future reference as follows:

D,RTR,SMP,and,FMGR's ID segments addresses are used by the 'AB' system command routine.

D.RTR and EDIT ID segment addresses are used in the track release algorithm, i.e., their tracks are not released on abnormal termination.

D,RTR and SMP ID segment addresses are used to exclude these two programs from BATCH timing.

The system does not charge BATCH time for D.RTR and SMP because if it did the time might run out and that would cause the program to be aborted. To abort either D.RTR or SMP would cause possible loss of a file or spool. Actually both D.RTR and SMP are heavily involved in I/O and so would not accumulate much BATCH time, in any case. For the same reason the AB command will not abort SMP or D.RTR, but instead the father is aborted or flagged. Next the time base start up routine is configured, the time base is started and the SET TIME message is started on its way. The privileged I/O system is configured and initialized, the disc protection option (\$PDSK,AB,1) is set up and halts are set into locations 2 and 3. These halts are designed to catch lost systems before they wander too far from home. At this point the system is ready to run and the start up routine transfers control to the dispatcher at \$XEQ. The dispatcher will find work to do (FMGR needs loading) and will take control.

In summary then, during start up the system:

- A. If RTE-III, sets up the system map
- B. Sets up the free memory poll
- C. If RTE-III, sets up the non-system maps
- D. Sets up swapping parameters
- E. Schedules FMGR and sets special flags for it — if present
- F. Sets up the privileged I/O system
- G. Sends SET TIME message
- H. Sets various initial parameters.



Dear SAM,

Here is a nice procedure for loading programs from a user supplied relocatable library which has not been generated into an RTE system. You might like to pass it on to your readers.

```
*RU,FMGR
:LG,n n-large enough to hold %PROG and entire user
      supplied library %LIBRY
:MR,%PROG %PROG-user's main program
:EX
*RU,LOADR,99
/LOADR:UNDEFINED EXTS
/LOADR: names of undefined externals
```

At this point the loader suspends.

```
*RU,FMGR
:MR,%LIBRY
:EX
*GO,LOADR,2,0,1 loader is rescheduled
/LOADR: PROG READY
/LOADR:$END
```

The loader will have scanned the LG area and loaded *only* those routines it needed to resolve the undefined externals, and *not* the entire library. The loader is rather smart in that it doesn't care what order the routines are presented in (see the attached examples). It will also pass through the LG area multiple times to resolve references made by library routines to other library routines. If the first library doesn't resolve all the references, the procedure can be repeated

with a second library. For the user who wants to run the loader from within FMGR a second copy of FMGR must be available to do the :MR,%LIBRY when the loader suspends.

Best Regards,

David Fitterman
United States Dept. of the Interior
Geological Survey
Denver, Colorado

* * *

Dear Dave,

Thanks very much for passing this information on to our readers. In looking back at the RTE Manual I find that this information was not presented except for a note next to a GO, Loader parameter.

Sam

Dear Sam,

I have tried several times to use the system parse routine from FORTRAN but have had no luck. Is this system routine FORTRAN callable?

* * *

YES, one of the nice things about RTE is that a number of very efficient routines that the operating system uses can also be called by the user.

The layout of the parsed buffer is shown below. In essence the system looks at everything between commas, parses this information, and puts the information into four words of a 33 word array. The first word of each group of four tells the type of information (0=NULL, 1=NUMERIC, 2=ASCII). The parsing routine accepts up to 8 parameters. The last word of the 33 word array contains the number of elements parsed. The following example shows how I use the routine.

```
DIMENSION IBUF(10), IPBUF(33), ILU(5), IREG(2),
EQUIVALENCE (X,IREG), (IB,IREG(2)) CALL
```

```
RMPAR (1LU)
```

```
:
```

```
:
```

```
X=REIO (1,ILU,IBUF,10)
```

```
CALL PARSE (IBUF,IB*2,IPBUF)
```

IBUF IS THE BUFFER TO BE PARSED

IPBUF IS THE PARSED BUFFER

IB*2 IS THE WORD COUNT TIMES TWO TO GIVE
THE# OF CHARACTERS

Note, that IB was equivalent to IREG(2); thus IB is the contents of the B-registers (the transmission log of the input request (positive number of words).

Sam

PARSE and INPRS

ON, ABCDE, 1,5, AB	ON	1	2		TYPE	
			0	N		
			SPACE	SPACE		VALUES
			SPACE	SPACE		
TYPES: 0 = NULL 1 = NUMERIC 2 = ASCII	ABCDE	1	2		TYPE	
			A	B		
			C	D		VALUES
			E	SPACE		
		1	1		TYPE	
			1			
			0		VALUES	
			0			
			•			
			•			
			•			
					TYPE	
		VALUES				
		33	5		No. ENTRIES	

RTEU/P-49

Listed below are the software parts and manuals which are shipped with RTE-II and RTE-III systems. Revision levels (A, B, C, D, etc.) or date codes (1543, 1546, etc.) are included for your reference.

SOFTWARE	RTE II	RTE III
Core Res Sys	92001-16012 C	92060-12003 B
Loader	92001-16002D	92060-16004 B
Multi Term Monitor	92001-16003 B	Same
Sys Library	92001-16005 1545	Same
DVP43 (Power Fail)	92001-16004 C	92060-16001 A
Autor (Auto Restart)	92001-16014 B	Same
Autor Source	92001-18014 —	Same
Spool Monitor	92002-12001 C	Same
Spool Program	92002-12002 C Option	92060-12001 A
Batch Monitor Libr	92002-16006 C Y13	Same
EDITR	92002-16010 C	Same
EDITOR	20805-60001 C	Not included
DVR00	29029-60001 1543	Same
DVR31 (7900 Disc)	29013-60001 C	Same
DVR32 (7905 Disc)	92060-16031 A	Same
7900 System Generator	92001-16013 C	92060-16029 A
Fixed Head Generator	92001-16018 1546	Not Supported
7905 System Generator	92001-16026 A	92060-16032 A
WHZAT	Not Available	92060-16006 B
Assembler	92060-12004 A	Same
XREF	92060-16028 A	Same
\$PVMP	Not applicable	92060-16035 A
FORTTRAN II	20875-60001 E	Same
	20875-60002 E	Same
	20875-60003 E	Same
	20875-60004 E	Same
	20875-60005 E	Same
FORTTRAN IV	24170-60001 C	Same
	24170-60002 C	
	24170-60003 C	Same
FORTTRAN IV (10K Area)	24177-60001 1442	Same
	24177-60002 1442	
ALGOL	24129-60001 C	Same
	24129-60002 C	Same
RELO SUBR LIBR-EAU	24151-60001 D	Same
RELO SUBR LIBR-FP	24248-60001 C	Same
FFP SUBR LIBR	12977-16001 1451	Same
FTN4 SUBR LIBRARY	24152-60001 C	Same
FTN II FORMATTER	24153-60001 C	Same

SOFTWARE	RTE II	RTE III
16K SIO MAG TAPE 9 TR	13022-60001 B	Same
16K SIO MAG TAPE 7 TR	13030-60001 B	Same
16K SIO PAPER TAPE RDR	20319-60001 A	Same
16K SIO PAPER TAPE PUNCH	20320-60001 A	Same
16K SIO SYSTEM DUMP	20335-60001 B	Same
16K SIO TTY (LP COMPAT)	24127-60001 C	Same
16K SIO 2767 LP	24166-60001 B	Same
16K SIO 2762/2615	24329-60001 A	Same
16K SIO 2607 LP	24347-16001 1346	Same
PUNCH/VERIFY	20312-60001 A	Same
24K SIO TTY (LP COMPAT)	29100-60017 A	Same
24K SIO SYSTEM DUMP	29100-60018 A	Same
24K SIO PAPER TAPE RDR	29100-60019 A	Same
24K SIO PAPER TAPE PUNCH	29100-60020 A	Same
24K SIO 2767 LP	29100-60022 A	Same
24K SIO MAG TAPE 9 TR	29100-60023 A	Same
24K SIO MAG TAPE 7 TR	29100-60049 A	Same
24K SIO 2762/2615	29100-60050 A	Same
Manuals		
RTE REFERENCE	92001-93001 —	92060-90004
BATCH/SPOOL MONITOR	92002-93001 — Option Y13	Same
ASSEMBLER	92060-90005 —	Same
FORTRAN II	02116-9015 —	Same
FORTRAN IV	5951-1321 —	Same
ALGOL	02116-9072 —	Same
RELO SUBR	02116-91780 —	Same
SIO DVR 2762/2615	02762-90002 —	Same
SIO DVR 2767 LP	12653-90004 —	Same
SIO DVR 2607 LP	12987-90006 —	Same
SIO DVR 7970 B/E 9 TR	13022-90010 —	Same
SIO DVR 7970B 7 TR	13029-90010 —	Same
SIO INTRO	5951-1369 —	Same
SIO SYS CONFIG.	5951-1374 —	Same
SIO SUBSYS	5951-1390 —	Same
ERROR MESSAGES	5951-1377 —	Same
DVR00	29029-95001 —	Same

Following is a list of the drivers, (with part numbers, and revision levels) available for RTE systems.

RTE DRIVERS

DRIVER	PART #	REV LEVEL	DESCRIPTION
DVR00	29029-60001	1543	
DVR11	29030-60001	B	2892 Card Reader
DVR12	29028-60002	A	2767 Line Printer
DVR12	92001-16020	1534	26XX Line Printers
DVR31	29013-60001	C	7900 DISC
DVR32	92060-16031	A	7905 DISC
DVR15	09601-16021	A	7261A Card Reader
DVR23	92202-16001	A	9 Track MT
DVR30	20747-60001	C	Fixed Head DISC
DVR24	25117-60499	D	7 Track MT
DVR72	09611-16005	A	6940 A/B Local & Remote
DVR62	02313-16001	A	2313B DVR
	29009-60001	C	2313B DVR
	29011-60001	E	R2313

DOS-IIIB MODULES

The Index below indicates the modules available for DOS-IIIB systems, HP 24307B, date code 1523.

This Index relates the names of the relocatable modules to the part numbers of the equivalent paper tapes and indicates the purpose of the modules. Modules not specifically designated for the 2100A/S or for the 21MX computers are to be used on either.

NAME	PART NUMBER	REV	DESCRIPTION
DISCH	24307-16069	1523	DISC MONITOR
\$EXMD	24307-16070	1523	EXEC MODULES
DVR00	20985-60001	1516	TTY-LIKE CONSOLE/TERMINAL
DVR01	20987-60001	1419	PAPER TAPE READER
DVR02	20989-60001	1419	PAPER TAPE PUNCH
DVR05	24157-60001	1419	TTY-LIKE CONSOLE
DVR15	24307-16017	1446	7261A MARK SENSE CARD READER
D2892	24272-60001	1419	2892B CARD READER (DVR11)
D2767	24168-60001	1419	2767A LINE PRINTER (DVR12)
D26XX	24307-16011	1446	DVR12 FOR 2607, 2610, 2614, 2613, 2618
DVR23	13024-60001	1446	7970B/E MAG TAPE
DVR26	24307-16018	1507	2762A/B AND 2615A CONSOLE
DVR30	24307-16073	1523	DISC BATCH DRIVER
DVR31	24156-60001	1419	7900/7901 DISC
DVR67	24341-60001	1419	12889A HI SPD SERIAL IF
DVR70	24307-16009	1446	DVR70 FOR 12618A SYNC INTERFACE
DVR71	24307-16013	1515	12967A SYNCH MODEM IF
DVR72	24350-16001	1523	12587B ASYNC DATA SET IF
DVR73	24377-16001	1523	12920A/B MULTIPLEXOR
DVR74	24307-16014	1515	12966A/12968A ASYNCH IF

EFMP	24309-60002	1523	EXT FILE MGR EXEC MODULES
	24309-60003	1523	EXT FILE MGR UTILITIES
JOBPR	24307-16071	1523	JOB PROCESSOR
RLODR	24307-16072	1523	RELOCATING/LINKING LOADER
ASMB	24307-16006	1419	2100/21MX ASSEMBLER
.FTN4	24170-60001	C	FORTTRAN IV COMPILER
	24170-60002	C	
	24170-60003	C	
FTN4	24177-60001	1442	FORTTRAN IV COMPILER (10K AREA)
	-60002	1442	
ALGOL	24129-60001	C	ALGOL COMPILER
	24129-60002	C	
XREF	24223-60001	1523	2100/21MX CROSS REF TABLE GEN
F4D.N	24152-60001	C	RELO SUBRLIBR FTN4
F2E.N	24151-60001	D	RELO SUBR LIBR (EAU)
F2F.N	24248-60001	B	RELO SUBR LIBR (FP)
FFP.N	12907-16001	A	2100A/S FFP SUBR LIBRARY
\$SETP	12907-16002	1350	2100A/S FFP SUBR \$SETP
ATD01	24381-16001	1503	ASYNCH TERMINAL DRIVER No. 1
ATD02	24307-16012	1442	ASYNCH TERMINALL DRIVER No. 2
PMT01	24307-16008	1438	PAGE MODE TERMINAL DRIVER No. 1
PMT02	24307-16016	1503	PAGE MODE TERMINAL DRIVER No. 2
SLC	24307-16010	1438	SYNCHRONOUS LINE CONTROL DRIVER
DVR33	24278-60001	1419	2100/21MX WCS DRIVER
MASMB	24332-60001	1419	2100A/S WCS MICRO ASSEMBLER
WCSUT	24333-60001	A	2100/21MX MICRO UTILITIES
MDEBUG	24334-60001	1419	2100A/S WCS MICRO DEBUG EDITR
XASMB	12978-16001	1437	21MX WCS MICRO ASSEMBLER
XDEBUG	12978-16002	1437	21MX WCS MICRO DEBUG EDITOR
FFP.X	12977-16001	1451	21MX FFP SUBR LIBRARY
XSETP	12977-16002	1451	21MX FFP SUBR \$SETP



The following tables list currently available customer manuals for Data Systems Division products. This list supersedes the list in the last issue of the **Communicator**.

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

Copies of manuals and updates can be obtained from your local Sales and Service office. The address and telephone number of the office nearest to you are listed in the back of all customer manuals.

Update packages are free of charge. If you require an update package only, send your request to:

Software/Publications Distribution
11000 Wolfe Road
Cupertino, Ca. 95014

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

A few words about documentation terms:

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

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

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

9600/9700 SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02005-90001	Real-Time Executive Software System	\$12.00	10/71	6/73
02313-93002	RTE 2313B Analog-Digital Interface Subsystem Operating and Service Manual	12.50	2/74	8/75
02320-93002	RTE System Driver DVR76 for HP 2320A Low Speed Data Acquisition Subsystem Programming and Operating Manual	1.00	8/74	
02321-93001	RTE System Driver DVR74 for HP 2321A Low Speed Data Acquisition Subsystem Programming and Operating Manual	1.00	8/74	
09600-93010	RTE System DVR11 for HP 2892A Card Reader Programming and Operating Manual	1.00	8/74	
09600-93015	91200A TV Interface Kit; Programming and Operating Manual	4.50	7/75	
09601-93007	RTE Device Subroutine for HP 5327A/B-H48 Counter	2.50	12/74	
09601-93009	RTE Device Subroutine for HP 5326A-H18 Counter	2.50	12/74	
09601-93014	RTE System Driver DVR15 Mark Sense Card Reader Programming and Operating Manual	1.00	8/74	1/75
09601-93015	RTE for 40-bit Output Register #12556B	1.00	10/74	
09603-93001	9603A/9604A Control System and Scientific Measurement Operating and Service Manual	7.50	6/75	12/75
09610-93003	ISA FORTRAN Extension Package Reference Manual	2.50	10/75	
09611-90009	9611A Operating 406 Industrial Measurement and Control System	.25	4/75	
09611-90010	HP 6940A/B Multiprogrammer Verification Manual	4.50	8/75	
12604-93002	RTE DVR40 for 12604B Data Source Interface	1.00	8/74	
12665-93001	RTE System Driver DVR65 for HP 12771A Computer Serial Interface Kit	1.00	8/74	
12989-99001	RTE System Driver DVA15 for Card Reader Punch Subsystem 2894	1.00	1/75	
24998-90001	DOS/RTE Relocatable Library Reference	10.00	3/76	*N
25117-93003	RTE System Driver DVR24 for HP 7970 Series Digital Magnetic Tape Unit	1.00	8/74	

9600/9700 SYSTEM MANUALS (Continued)

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
29003-93001	RTE System Driver DVR66 for HP 12772A Coupler Modem Interface Kit Programming and Operating Manual	1.00	8/74	
29003-93003	RTE System Driver DVR66 for HP 12770A Coupler Serial Interface Kit Programming and Operating Manual	1.00	8/74	
29009-93001	RTE System Driver DVR62 for HP 23138 Subsystem	2.50	8/74	
29013-90001	DVR31 RTE Moving Head Driver	10.00	2/73	
29014-90001	Moving Head Real-Time System Generator	20.00	4/72	
29015-90001	Fixed Head Real-Time System Generator	15.00	4/72	
29016-90002	RTE Scheduler	50.00	9/72	
29016-90003	Real-Time Input/Output Control	50.00	12/73	
29022-90001	Real-Time Relocating Loader	10.00	6/73	
29028-95001	RTE HP 2610A/2614A Line Printer Driver	1.50	8/73	
29029-91001	Real-Time Executive Multiple-Device System Control Device (DVR00) Program Listing	10.00	9/72	
29029-95001	Real-Time Executive System Driver DVR00 for Multiple Device System Control Small Programs Manual	1.50	11/75	
29033-98000	Real-Time Executive-File Manager System	10.00	3/73	
29100-93001	RTE System Driver DVR40 (29100-60041) for HP 12604B Data Source Interface Programming and Operating Manual	1.00	8/74	10/74
29100-93003	RTE System Driver DVR61 for HP 6940A, 6941A Bidirectional Multiprogrammer Programming and Operating Manual	3.00	8/74	
29101-93001	RTE Core-Based Software System Users Manual	10.00	1/76	
29102-93001	RTE BASIC Software System Programming and Operating Manual	10.00	3/74	8/75
29103-93001	RTE System Cross Loader; Programming and Operating Manual	2.50	3/75	11/75
91060-93005	RTE Driver for X-Y Display Storage Subsystem (HP Model 1331C-016) Programming and Operating Manual	1.00	8/74	
91062-93003	Real-Time Executive System Driver for DVM/Scanner Subsystem	9.00	8/74	
92001-93001	RTE-II Software System Programming and Operating Manual	10.00	3/76	*R
92060-90004	RTE-III Software System Programming and Operating Manual	12.00	3/76	*R
92060-90005	RTE Assembler Reference Manual	7.00	1/76	*R
92060-90009	RTE-III General Information Manual	4.00	2/76	*N
92060-90010	RTE Batch/Spool Monitor and Operating System Pocket Guide	3.00	10/75	
92060-90012	RTE-III: A Guide for New Users	6.50	10/75	
92060-90013	Batch-Spool Monitor Reference Manual	9.50	12/75	3/76
92060-90014	RTE Interactive Editor Reference Manual	6.00	3/76	*R
92060-90016	Multi-User Real-Time BASIC Reference Manual	12.00	10/75	4/76
92200-93001	RTE System Driver DVR12 for HP 2607A Line Printer Programming and Operating Manual	1.00	3/74	
92200-93005	Real-Time Executive Operating System Drivers and Device Subroutine Manual	\$ 5.00	3/76	*R
92202-93001	RTE System Driver DVR23 for HP 7970 Series Digital Mag Tape Units Programming and Operating Manual	1.00	8/74	
93005-93005	Thermal Line Printer Subsystem for Driver DVR00 (RTE)	2.50	12/74	
93513-90002	RTE System Driver DVA 76-DVR40 for 2801 Quartz Thermometer System	1.50	4/75	

*R = Revised

*N = New

SOFTWARE INPUT/OUTPUT SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02116-91760	Teleprinter Driver (LP Compatible) Manual	\$ 1.00	8/73	
02762-90002	HP 2762A Terminal Printer Driver	1.00	5/73	
02892-90003	HP 2892A Card Reader Driver	1.50	6/72	
12602-90022	Mark Sense Card Reader Drivers	1.00	6/70	
12653-90004	HP 2767 Line Printer Driver	1.00	9/70	1/73
12845-90005	HP 2610A/2614A Line Printer Driver	1.00	2/74	
12987-90006	HP 2607 Line Printer Driver	5.00	1/73	
13022-90010	HP 7970 Magnetic Tape Unit Driver	1.00	2/72	
13029-90010	Magnetic Tape Driver (7-Track)	1.00	2/72	
5950-9276	SIO Drum-Disc	1.00	2/70	
5951-1374	Software Input/Output System Configuration	1.00	7/74	
5951-1390	Subsystem Operation	2.00	2/76	*R

BASIC CONTROL SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02022-90014	Magnetic Tape Reformatting System Support Utilities	\$ 1.50	1/74	
02100-90129	HP 2100 Microassembler Coding Form	5.00		
02100-90140	Decimal String Arithmetic Routines	6.50	10/73	
02108-90008	Microprogramming 21MX Computers Reference Manual	5.00	8/74	10/4
02116-9017	Basic Control System Manual	8.50	12/71	
02116-9072	ALGOL for HP 2000 Computers Reference Manual	10.00	2/76	*R
02116-91751	Prepare Tape System	2.50	8/74	
02116-91752	Magnetic Tape System	6.00	6/71	
02116-91780	2100 Series Relocatable Subroutines	11.00	12/74	
02762-90003	HP 2762A Terminal Printer Driver	1.00	5/73	
02892-90004	HP 2892A Card Reader Driver	1.50	6/72	
12602-90021	Mark Sense Drivers	1.00	6/70	
12653-90005	HP 2767 Line Printer Driver	1.00	10/70	
12845-90004	HP 2610A/2614A Line Printer Driver	1.00	6/72	
12987-90008	HP 2607 Line Printer Driver	5.00	12/73	
13023-90010	HP 7970 Magnetic Tape Unit Driver	1.00	5/74	
13026-90010	Magnetic Tape Driver (7-Track without DMA)	1.00	5/71	6/72
13027-90010	Magnetic Tape Driver (7-Track with DMA)	1.00	5/71	6/72
5951-1371	HP 2100 Front Panel Procedures	1.00	8/73	
5951-1376	Basic Binary Loader/Disc Loader, Basic Moving-Head Disc Loader	1.00	4/74	
5951-1391	Basic Control System	1.50	10/74	
5951-1392	Magnetic Tape System	1.00	7/71	

*R = Revised

DISC OPERATING SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02767-90007	DOS/RTE 2767 Line Printer Driver	\$ 1.00	12/70	
12560-90023	DOS RTE and BCS Calcomp Plotter Drivers	1.50	10/75	
12602-90023	DOS/RTE Mark Sense Drivers Kit 12602B	1.00	8/70	
12908-90004	HP 12908 Writable Control Store Driver	1.00	2/75	
24307-90006	DOS-III Reference Manual	20.00	1/76	3/76
24307-90012	DOS-III Data Communications Drivers	7.50	8/75	11/75
24307-90018	DOS-III Pocket Guide	3.50	12/75	
24307-90022	DOS-III Terminal Printer Driver	1.00	1/75	
24307-90073	DOS-III Standard Drivers	6.00	1/75	
24376-90001	IMAGE/2000 Data Base Management System Reference Manual	11.00	8/75	
5951-1366	Cross Reference Table Generator	1.00	8/74	
5951-1381	DOS-M/2000C Timeshared BASIC File Handler	1.00	5/71	
5951-1394	2000C File Interface for DOS-M	1.00	6/71	

LANGUAGE MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02116-9014	HP Assembler Manual	\$ 6.50	8/75	
02116-9015	HP FORTRAN Manual	5.00	3/74	
02116-9016	Symbolic Editor	4.50	2/74	
02116-9072	ALGOL Reference Manual	10.00	2/76	
12907-90010	Implementing the HP 2100 Fast FORTRAN Processor	5.00	11/74	
24307-90014	DOS III Assembler Reference Manual	8.00	7/74	11/75
92060-90005	RTE Assembler Reference Manual	7.00	5/75	
5951-1321	HP FORTRAN IV Reference Manual	6.00	12/75	

*R = Revised

The schedule for customer training courses on Data Systems Division products has been expanded to include courses offered at our European training centers. Listed below are courses offered in the U.S. and in Europe during the period May through August 1976.

You can also obtain a copy of the training schedule from your local HP sales office. A European course schedule is available through the sales offices in Europe; a U.S. schedule through U.S. sales offices.

*Prices quoted are for courses at the two U.S. training centers only. For prices of courses at European training centers please consult your local HP Sales Office.

Registration

Requests for enrollment in any of the above courses should be made through your local HP representative. He will supply the Training Registrar at the appropriate location

with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the Training Course, time of class, location and accommodations reserved.

Accommodations

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

Cancellations

In the event you are unable to attend a class for which you are registered please notify the Training Center Registrar immediately in order that we may offer your seat to another student.

Training Center Addresses

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

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

Boblingen
Kundenschulung
Herrenbergerstrasse 110
D-7030 Böblingen, Wurttemberg
Tel: (07031) 667-1
Telex: 07265739
Cable: HEPAG

Winnersh
King Street Lane
GB-Winnersh, Wokingham
Berks RG11 5 AR.
Tel: Wokingham 784774
Cable: Hewpie London
Telex: 847178 9

Grenoble
5, avenue Raymond-Chanas
38320 Eybens
Tel: (76) 25-81-41
Telex: 980124

Milan
Via Amerigo Vespucci, 2
1-20124 Milan
Tel: (2) 62 51
Cable: HEWPACKIT Milano
Telex: 32046

Madrid
Jerez No 3
E-Madrid 16
Tel: (1) 458 26 00
Telex: 23515 hpe

Stockholm
Enighetsvägen 1-3, Fack
S-161 20 Bromma 20
Tel: (08) 730 05 50
Cable: MEASUREMENTS Stockholm
Telex: 10721



COURSE NUMBER	TITLE		TRAINING COURSE RATES AND CENTER LOCATIONS										AMSTERDAM/ BRU.
	LENGTH	PRICE	CUPERTINO	ROCKVILLE	BÖBLINGEN	WINNERSH	GRENOBLE	MILAN	MADRID	STOCKHOLM			
22940A	2100 MAINT. 10 days	\$1000	5/3 6/21 8/2	5/10 8/23									
22941A	21MX MAINT. 5 days	\$ 500	5/17 7/12	8/9			5/17 (English)						
22942A	7900 MAINT. 5 days	\$ 500	5/17 7/19 8/23				5/24 7/12 (English)						
22943A	7970 B/E MAINT. 5 days	\$ 500	5/24 8/16				5/31 7/19 (English)						
22950A	2100 SER. ASSM. 5 days	\$ 500	5/24 6/21 7/19 8/16	5/3 5/24 6/7 6/28 7/26 8/16									
22952A	DOS III B 5 days	\$ 500	7/12										
22952B	DOS III B 5 days	\$ 500			8/9		7/5		5/17	5/31	8/23		
22953A	2100 IMAGE 3 days	\$ 300	7/21										
22959A	ASSEMBLER/21MX 5 days	\$ 500			5/3 8/2	5/10 7/5	6/21	6/7	5/10		5/10		
22960A	21MX MIC. PROG 5 days	\$ 500	6/7 7/26										
22965B	RTE-II/III 10 days (Course includes: RTE-II/III operat- ing system, batch spool monitor and file manager)	\$1000	5/10 6/7 6/14 7/12 7/26 8/9 8/16	5/17 6/14 7/12 8/2 8/23	* { 5/10 } * { 5/17 } * { 6/21 } * { 6/28 } * { 8/30 } * { 9/6 }	* { 7/12 } * { 7/19 }	* { 6/28 } * { 7/19 }	* { 6/21 } * { 7/5 }	* { 5/31 } * { 6/7 }	* { 5/3 } * { 5/10 } * { 8/30 } * { 9/6 }	* { 5/31 } * { 6/7 }		
22968A	MEASUREMENT & CONTROL 2 days	\$ 200	5/27 6/28 8/9 8/30				5/31 (Francais)						
22969A	DISTB. SYS. 5 days	\$ 500	5/17 8/2	5/31	8/16	8/9	5/17						
22978	TCS 2 days	\$ 200	7/19										
22979A	REAL TIME BASIC 3 days	\$ 300	5/24 6/30	8/11 9/1									

*Note: Dates within brackets are starting dates for week 1 and week 2 of the RTE course. In some cases there is a break between the two weeks of the class.

**HEWLETT-PACKARD
COMPUTER SYSTEMS COMMUNICATOR ORDER FORM**

Please Print:

Name _____ Title _____
Company _____
Street _____
City _____ State _____ Zip Code _____
Country _____

☐ MAIL ORDER SUBSCRIPTIONS

BASE SUBSCRIPTION \$ _____

_____ ADDITIONAL SUBSCRIPTION(S) \$ _____
No.

TOTAL AMOUNT ENCLOSED \$ _____

☐ SOFTWARE SERVICE CONTRACT SUBSCRIPTIONS

BASE SUBSCRIPTION (NO ADDITIONAL CHARGE) NAC

_____ ADDITIONAL SUBSCRIPTION(S) \$ _____
No.

TOTAL AMOUNT YOU WILL BE BILLED \$ _____

FOR HP USE ONLY

SUPPORT OFFICE NUMBER _____ ORDER DATE _____
APPROVED BY _____ C.E. NUMBER _____
SERVICE CONTRACT NUMBER _____ EXPIRATION DATE _____
AUTHORIZED TOTAL NUMBER OF SUBSCRIPTIONS _____
CUSTOMER'S HP OPERATING SYSTEM _____

Printed 4/76

TABLE OF PRORATED \$ AMOUNT DUE PER ADDITIONAL SUBSCRIPTION

(Use only for ordering ADDITIONAL SUBSCRIPTION(S) against an existing Software Service Contract)

Months Remaining in Service Contract												
	1	2	3	4	5	6	7	8	9	10	11	12
Cost of Each ADDITIONAL SUBSCRIPTION	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	11.00	12.00

INSTRUCTIONS FOR ORDERING COMMUNICATOR

All Hewlett-Packard customers with Software Service Contracts are entitled to one BASE SUBSCRIPTION (1 copy per issue) at no additional charge. These customers may also buy ADDITIONAL SUBSCRIPTIONS whose purchase price is to be prorated against the remaining life of their Software Service Contract.

Customers who do not have Software Service Contracts may purchase Mail-Order Subscriptions through HP's Direct Mail Order System.

A. MAIL-ORDER SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
 2. Compute amount due:
 - a) Annual Base Subscription (6 issues) \$ 48.00
 - b) _____ Additional Subscriptions*
@ \$12.00 ea. \$ _____

 - c) Total Order Amount (a + b) \$ _____
 - d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Mail check or bank draft with ORDER FORM to:

HEWLETT-PACKARD CO.
MAIL ORDER DEPARTMENT
P.O. DRAWER #20
MOUNTAIN VIEW, CA. 94043
U.S.A.

B. SOFTWARE SERVICE CONTRACT SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
 2. Compute amount due: (BASE SUBSCRIPTION is at no additional charge.)
 - a) Annual Base Subscription (6 issues) \$ 0.00
 - b) _____ Additional Subscriptions*
\$ _____

Prorate the dollar amount to make the ADDITIONAL SUBSCRIPTIONS EXPIRE WITH YOUR Software Service Contract. (SEE TABLE)

 - c) Total Order Amount (a + b) \$ _____
 - d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Forward ORDER FORM to your local HP Customer Engineering Representative. Your order will be approved and forwarded to the appropriate department. You will be billed for any ADDITIONAL SUBSCRIPTIONS by your local HP office.

C. SPECIAL INSTRUCTIONS FOR INTERNATIONAL CUSTOMERS

1. International customers who do not have Software Service Contracts are encouraged to use HP's Direct Mail Order System by remitting a bank draft in U.S. dollars according to the ordering procedures outlined in Instruction A above. Optionally, international customers may purchase the **Communicator** through their local HP Sales and Service Office. The customer should contact his HP Office for the subscription prices in the currency of his country, then complete the Order Form and forward it together with payment to his local HP Customer Engineering Department.

1. International customers with Software Service Contracts should follow the ordering procedure outlined in Instruction B above. If the customer wishes to purchase ADDITIONAL SUBSCRIPTIONS, he should contact the local HP Office for the subscription price in the currency of his country, then submit the ORDER FORM. The customer will be billed for ADDITIONAL SUBSCRIPTIONS by his local HP Office.

*All ADDITIONAL SUBSCRIPTIONS will be sent to the same name and address as the BASE SUBSCRIPTION.



CORPORATE PARTS CENTER

Direct Mail Parts and Supplies Order Form

SHIP TO:

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

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

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

HEWLETT-PACKARD COMPANY

Mail Order Department

P.O. Drawer #20

Mountain View, CA 94043

Phone: (415) 968-9200

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

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

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

direct mail order form

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

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

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



Computer Systems **Communicator**

Subscription Service Manager

Hewlett-Packard Company

Mail Order Department

P.O. Drawer No. 20

Mountain View, California 94043

U.S.A.

May 1976

Bulk Rate

U.S. Postage

PAID

Sunnyvale, CA. 94086

Permit No. 541

Address Correction Requested
Forwarding and Return Postage Guaranteed