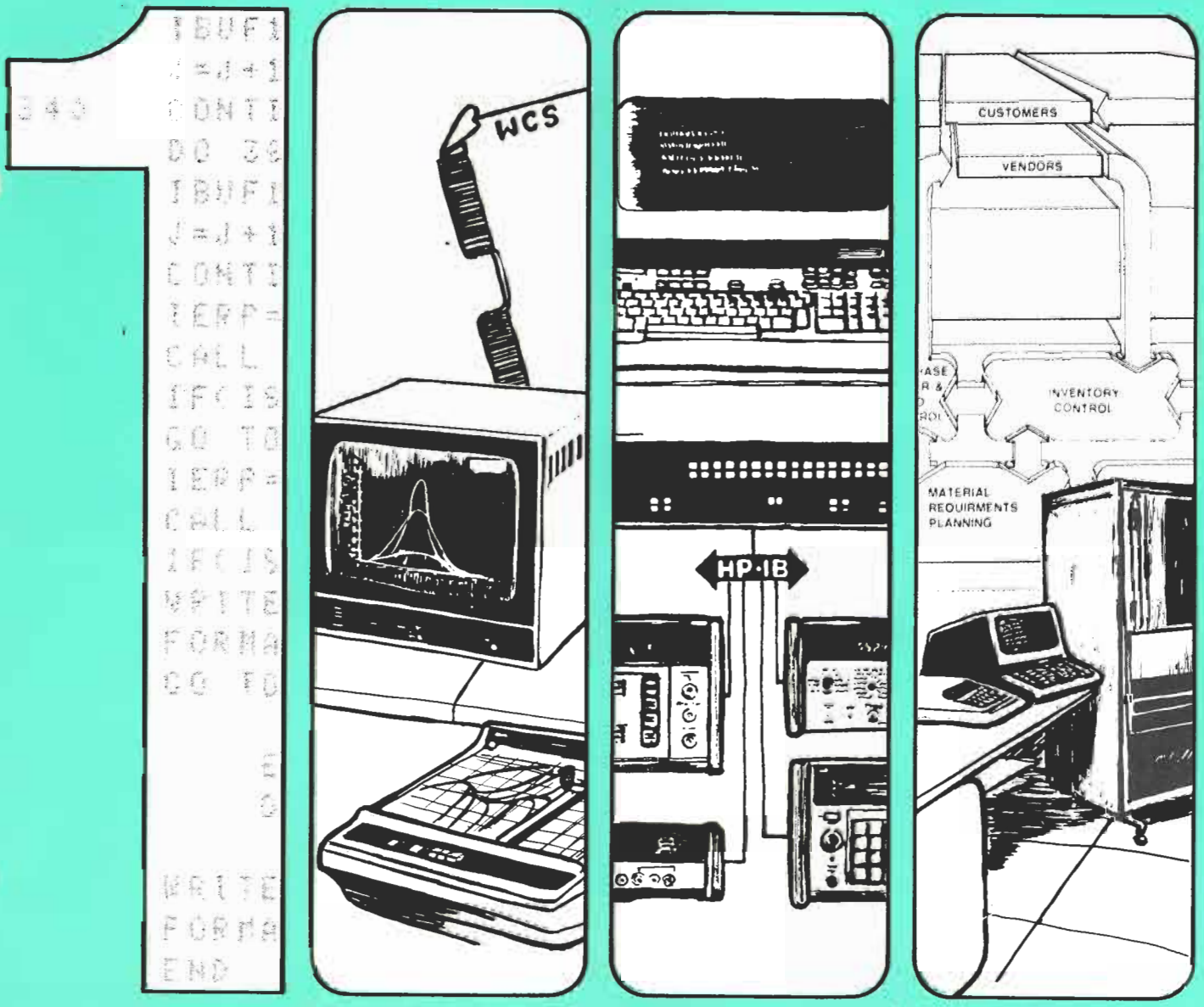
HEWLETT hp PACKARD

Computer Systems

COMMUNICATOR



issue no. 15

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

EDITOR'S NOTE

We Heard You:

Thanks to your inputs we included an Index to articles in past COMMUNICATORS in our last issue. Now, you can use that Index to determine which back issues of the COMMUNICATOR contain information you would like to have? The back issues available are #'s 4,5,8,9,11,12,13 and 14 — they can be ordered by filling out the order form in the back of this issue. Cost of back issues is a nominal $10.00.

We Hear You:

The series of articles "KNOW YOUR RTE" has appeared regularly in the COMMUNICATOR. Due to much positive feedback, we have published a collection of these articles, parts one through six. Please refer to this issue's "KNOW YOUR RTE" article for more information on this collection, along with in-depth RTE material.

We Want to Hear From You:

Have any software you feel other customers might benefit from? If so, become one of the many who have added their software to LOCUS (Library of Contributed User Software). You will find in this issue all the information and forms (perforated for easy removal) necessary to become a contributor. Remember, all first time entries are awarded a beautiful engraved Wall Plaque with the contributor's name and a program name disc for each accepted program. Also, for each program accepted, you can receive a complimentary program of your choice from the existing LOCUS.

Interested in receiving a free LOCUS catalog (list price $15)? Send us an article dealing with some technical aspect of the HP 1000 Computer System (hardware or software) that you feel may benefit our other readers. We are soliciting articles for issue #'s 16, 17, and 18 of the COMMUNICATOR 1000. For each issue we will select two articles to publish, the authors of which will receive a free LOCUS catalog or a program of your choice from the LOCUS catalog (not to exceed $20). The articles should be no longer than 3 typed pages not including listings (if any). Mail your inputs to the address given below. This is a great opportunity to let others benefit from your experience. Hope to be hearing from you.

What You'll Hear From Us:

In this issue . . . a Systems Maintenance UTility (SMUT), hints on Microprogramming techniques, Error Correcting Memory . . . and much, much more.

We at Hewlett-Packard are doing our best to keep you informed about the HP 1000.

Please address any correspondence to:

EDITOR

COMPUTER SYSTEMS/COMMUNICATOR 1000
HP Data Systems Division
11000 Wolfe Road
Cupertino, CA. 95014

CONTENTS

MAKING MICROPROGRAMS INTERRUPTIBLE

Bill Elmore/DSD

Why should a microprogram be interruptible?

Should your microprogram be interruptible?

How to do it!!

Examples

When using the power of microprogramming to help you improve your system's performance, it is important to understand the effect of the microprogram upon its environment. Particularly under an RTE operating system, it becomes essential that the programmer be aware of the needs of the operating system with regard to his microprogram.

Once a microprogram starts executing, it has complete control of the CPU and cannot be interrupted by any device unless the microprogram itself checks for interrupts. (In fact, even the HALT button will have no effect unless the microprogram specifically checks for a HALT condition.) It is not desirable to hold off interrupts for very long; therefore you must decide how long your microprograms can be allowed to execute before testing for an interrupt. In making this decision, consider the impact that a long non-interruptible microprogram can have in the RTE environment.

Probably the most catastrophic of all interrupts is the power fail interrupt. After a power fail, the user has 500 microseconds to save the state of the machine before the CPU is actually shut down. Therefore, the user should ensure that his microprogram is interruptible at least every:

(500 — Time for Power Fail Routine to Execute) usecs.

Another consideration may be a very fast device that needs to be serviced quickly. In this case, the microprogram should be interruptible often enough to ensure no loss of data from the device. Under RTE, the time base generator (TBG) interrupts every 10 msec. It is up to the user to determine whether holding off TBG ticks is critical to his application.

A general rule of thumb is that microprograms should be interruptible at least every 45 usec. (This is the execution time of Floating Point Divide, the slowest non-interruptible HP instruction.) If your microprogram is written within these constraints, you can feel confident that you are "safe" with any HP software. Of course, your special application may warrant that you check for interrupts more often than 45 usec.

Now that you are able to determine when to make your microprograms interruptible, how do you actually accomplish this task?

In a 21MX M-Series Computer (2105/2108/2112), one implements interruptibility as follows: On interrupt, ensure that P is pointing to the macroinstruction that you desire to execute upon return from the interrupt. Then you must save all values that you will need upon return from the interrupt. Since interrupt handlers should save A,B,X,Y,S,O, & E, these are good places to store your intermediate data. Data may also be saved in main memory, to be retrieved upon re-entry to the microprogram. It may also be necessary to set a flag to indicate whether this is a re-entry or first entry to the microprogram. To process the interrupt, you must then execute a RTN instruction (if you are not in a micro-subroutine) or a JMP to control memory location 004. (This is the location of the interrupt handling microcode in the M-Series computer.) Figure 1 is an example interruptible microprogram for an M-Series computer.

In an E-Series Computer, it is necessary to set P to one instruction past the instruction to be executed on return from interrupt. To process the interrupt, a JMP to control memory location 006 is recommended. (You can still execute a RTN if you are certain that you will never be in a micro-subroutine. . . remember, the E-Series computer has three levels of subroutining.) Otherwise the technique for handling interrupts is the same as the M-Series computers. Figure 2 is an example interruptible microprogram for an E-Series computer.

In many cases, it will not even be necessary to make your microprogram interruptible, but interruptibility is something you should be concerned with every time you write a microprogram.

By using some of the guidelines in this article and, of course, a complete understanding of your particular application, you will be better able to use microprogramming to increase your system's performance.

Figure 1.

```
                    MICMX,L
                    $CODE='M2.1,REPLACE
                              ORG  6000B
                    ************************************************************
                    *                                                          *
                    * CALLING SEQUENCE                                         *
                    *       LDA NMRR     + NUMBER OF SORT ELEMENTS             *
                    *       LDB TABLE    ADDRESS OF FIRST ELEMENT              *
                    *       CLE          E=(0=INITIAL ENTRY,                   *
                    *                       1=RETURN FROM INTERRUPT)           *
                    *       OCT 105600   INVOKE SORT MICROPROGRAM              *
                    *                                                          *
                    ************************************************************
06000  321 101030               JMP                    SORT      SAVE ENT POINTS
                                ALGN
                    ******************************************
                    * SAVE P (NEXT INSTRUCTION ADDRESS) IN S11 *
                    ******************************************
06020  017 175517   SORT                    S11  P              S11 = NEXT
                    ************************************          INSTR ADDR
                    * RETURNING FROM INTERRUPT ? (E=1?) *
                    ************************************
06021  324 142771               JMP  CNDX E            INTRTN    YES, USE INTRTN
                    ******************************************
                                          .
                                          .
                                          .
                                BODY OF MICROPROGRAM
                                          .
                                          .
                                          .
                    ********************
                    * ANY INTERRUPTS ? *
                    ********************
06053  326 043331   INTCHK   JMP  CNDX NHOI       ENDCHK       NO, CHK PASS
                    ******************************************
                    * SAVE P (NEXT I ADDRESS) IN X (X=P)     *
                    * SET INTERRUPT RETURN INDICATOR (E=1)   *
                    * FIX P (P=S11-1), EXIT                  *
                    ******************************************
06054  017 175617   INTEXIT                  X    P          X = NEXT I ADDR
06055  347 000417            IMM       LOW  IR   300B       IR(7,6)=(1,1)=
06056  207 165736            ASG  RTN  DEC  P    S11         CCE, FIX P,
                    *                                         EXIT
                    ************************************************
                    * RESTORE ADDRESS OF NEXT I IN P (P=X)         *
                    * RESTORE ADDRESS OF NEXT J IN S4 (S4=P+Y)     *
                    * RESTORE NUMBER OF COMPARES IN S3 (S3=(B+A)-S4) *
                    ************************************************
06057  017 171717   INTRTN                      P    X          P = NEXT I ADDR
06060  017 172157                              L    Y
06061  004 175157                   ADD  S4    P               S4 = NEXT J ADDR
06062  017 126157                              L    A
06063  004 125117                   ADD  S3    B               S3 = B+A
06064  017 146157                              L    S4
06065  003 045117                   SUB  S3    S3              S3 = (B+A)-S4 =
                    *                                         COMPARES
                    ******************************************
                    * MORE COMPARES ? (S3=S3-1, S3 NOT =0?) *
                    ******************************************
06066  007 145117   ENDCHK             DEC  S3   S3           MORE COMPARES ?
06067  320 001531            JMP  CNDX TBZ  RJS  COMPARE      YES,DO NEXT
```

P ▸ one past OCT 105600

E = 1 indicates return from interrupt

less than 45 μsec

CHECK for interrupts

P ▸ OCT 105600

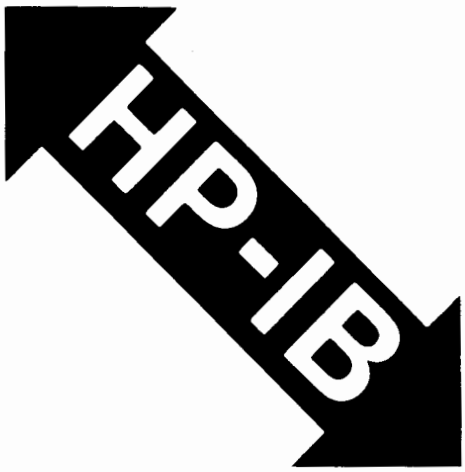
restore intermediate values upon return from interrupt

```
0002 RTE MICRO-ASSEMBLER REV.A 760805

                    MICMXE,L
                    $CODE='M2.1E,REPLACE
                            ORG  34000B
                    ************************************************************
                    *                                                          *
                    * CALLING SEQUENCE                                         *
                    *       LDA NMBR     + NUMBER OF SORT ELEMENTS             *
                    *       LDB TABLE    ADDRESS OF FIRST ELEMENT              *
                    *       CLE          E=(0=INITIAL ENTRY,                   *
                    *                       1=RETURN FROM INTERRUPT)           *
                    *       OCT 105600   INVOKE SORT MICROPROGRAM              *
                    *                                                          *
                    ************************************************************
                    HORI    EQU  6B
34000  327 001007           JMP                  SORT      SAVE ENT POINTS
                            ALGN
                    ********************************************
                    * SAVE M (NEXT INSTRUCTION ADDRESS) IN S11 *
                    ********************************************                 M ⏵ one past OCT 105600
34020  010 033507   SORT                 S11  M          S11 = NEXT             P ⏵ two past OCT 105600
                    ************************************    INSTR ADDR
                    * RETURNING FROM INTERRUPT ? (E=1?) *
                    ************************************
34021  334 103042           JMP  CNDX E          INTRTN    YES, USE INTRTN   E = 1 indicates return
                    ****************************************                        from interrupt.
                                          .
                                          .
                                          .
                                BODY OF MICROPROGRAM
                                          .
                                          .
                                          .
                    ********************
                    * ANY INTERRUPTS ? *
                    ********************
34054  323 143442   INTCHK   JMP  CNDX HOI  RJS  ENDCHK      NO, CHK PASS      CHECK for interrupts
                    ********************************************
                    * SAVE P (NEXT I ADDRESS) IN X (X=P)       *
                    * SET INTERRUPT RETURN INDICATOR (E=1)     *
                    * FIX P (P=S11)
                    * JMP TO HORI (BASE SET INTERRUPT CODE)    *
                    ********************************************
34055  010 075607   INTEXIT                X    P          X = NEXT I ADDR
34056  342 000607            IMM       LOW  IRCM 200B      IR(9-6)=1110=ELA
34057  011 136761                 SRG1 ONE                 CCE
34060  010 065707                      P    S11        CCE,RSS, FIX P,      P ⏵ one past
34061  320 000307            JMP                 HORI      JMP TO BASE SET      OCT 105600
                    *                                       INTERRUPT CODE
                    ************************************************
                    * RESTORE ADDRESS OF NEXT I IN P (P=X)         *
                    * RESTORE ADDRESS OF NEXT J IN S4 (S4=P+Y)     *
                    * RESTORE NUMBER OF COMPARES IN S3 (S3=B+A-S4) *
                    ************************************************
```

```
34062  010 071707   INTRTN                P     X          P = NEXT I ADDR
34063  010 072507                         L     Y
34064  003 075147                ADD     S4    P          S4 = NEXT J ADDR
34065  010 006507                         L     A
34066  003 011107                ADD     S3    R          S3 = B+A
34067  010 046507                         L     S4
34070  004 145107                SUB     S3    S3         S3 = (B+A)-S4 =
                                                            COMPARES
                    *
                    ****************************************
                    * MORE COMPARES ? (S3=S3-1, S3 NOT =0?) *
                    ****************************************
34071  000 045107   ENDCHK               DEC   S3    S3         MORE COMPARES ?
34072  320 041542              JMP  CNDX ALZ   RJS   COMPARE      YES, DO NEXT
```

restore intermediate values upon return from interrupt



HP-IB TREKIE ARTICLE #5

DATA SETTLING TIMES & DAV GLITCH

Larry W. Smith/DSD

DATA SETTLING TIME

The standard (I.E. IEEE 488-1975) specifies a data settling time which may be either 2 micro-seconds or 500 nano-seconds, depending upon the type of drivers used on the data lines. Simulation results indicate that a rising signal always requires more settling time than a falling signal, so discussion of the worst-case times will assume low-to-high transitions only.

If the line drivers are of the open collector type, the standard requires 2 micro-seconds settling delay. Calculations in Article #4 indicate that the open collector rise time will be 814 nsec for the worst-case condition allowed by the standard (that is, two meters of cable per device). Thus, the 2 micro-seconds requirement is certainly sufficient and perhaps longer than necessary.

If the drivers are tri-state type, the standard permits the settling time to be reduced to 500 nsec. Simulation of several bus configurations indicates that 500 nsec is sufficient. In addition, if the total cable length is limited to one meter per device and the maximum number of devices is fifteen, the simulation results show that the signals will always settle in less than 500 nsec. Based on these findings, it seems reasonable to permit 350 nsec settling time so that the data transfer rate may be maximized on systems which can limit the cable length.

DAV GLITCH

During the simulations, it was found that two low-to-high transitions can occur on the rising edge of a tri-state signal. On the data lines this does not present a problem, since the signal is settled before DAV is asserted. However, if the dual transition occurs on DAV, it can cause errors with an acceptor which is ready for successive messages within about 100 nsec. That is, if the acceptor is fast enough to send RFD within 100 nsec after DAV is removed, the glitch can appear to be a valid assertion of DAV and thus initiate a new handshake cycle. The result would be the acceptance of invalid commands (possibly by several devices) if ATN is true or the receipt of false data by a listener. Possible methods of preventing this situation include:

a. Reduce the output impedance of the tri-state driver so the initial signal level is higher and subsequent signal reflections at the driver are reduced. As a result, the driver would be required to source more than the 5.2 MA currently specified by the standard. This would have an adverse effect upon existing HP-IB devices, since TTL buffers do not normally have a capability greater than 5.2 MA.

b. Require the DAV line to be driven by an open collector driver. This would probably require a design change in many existing devices and would also result in a significant reduction in bus data rate.

c. Require a DAV settling time in the acceptor so that a DAV assertion would not be considered valid until the settling time was satisfied.

d. Restrict the bus to a linear topology and require all receivers on the DAV line to have SCHMITT-TRIGGER inputs with 1.1 volt maximum negative-going threshold. This might solve the problem, but the versatility would be reduced and existing devices would be effected. Settling time would not seriously affect the bus performance. However, such a redesign would certainly effect existing devices unless they already include this feature by virtue of their internal delays.

KNOW YOUR RTE — PART 9

By Mr. RTE

This series of articles has appeared regularly in the *Computer Systems Communicator*. Because of its popularity and value as reference material, we are publishing a collection of these articles for those who are involved with the RTE environment.

This group of articles consists of "Know Your RTE" parts one through six, which were published in issues seven through twelve. They cover the following subjects:

Part 1 The List Processor, $LIST
Part 2 System Start-Up or "Bootstrap"
Part 3 Operator Requests through System Console
Part 4 Operator Requests Continued ($MESS)
Part 5 I/O Requests
Part 6 Time Keeping

The collection of all six articles is available in Xerox form (part number 5955-3205) may be obtained from your local HP sales office free of charge. If they have run out, have them order more.

In this issue, we will discuss resource numbers, Class I/O, LU Lock and LU switching. The reader is referred to the article, (A Real-Time Executive System with Multi-Terminal and Batch/Spool Capabilities) by ***George A. Anzinger*** and ***Adele M. Gadol***, which appeared in the December, 1975 issue of the HP Journal. A reprint of this article, along with others, is also available as part number 5952-9929 from your local sales office. It is FREE. This article contains the best (i.e., most understandable) description of Class I/O that we have seen.

CLASS I/O (INITIALIZATION)

Class I/O initiation is almost the same as standard I/O which we covered in an earlier article in this series. The differences are:

a. That a class number may need to be allocated

b. Class I/O is disallowed on discs

c. An extra word is requested in the buffer request (to hold the class number)

d. The read request does not need a buffer move.

In connection with the class number allocation check, the end of the class queue is found. The word at the end of the queue is formatted as per Figure 1 and is checked for conformance to the class security code. This security code is the least 5-bits of the ID segment number of the program which allocated the class.

If everything is right, after the request is queued, the number of pending requests counter for the class is incremented.

Figure 1.

```
CLASS I/O QUEUE FORMAT AND ITS USE

  THE CLASS QUEUE CAN BE IN FOUR DIFFERENT STATES.
    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 1:  CLASS DEALLOCATED, AVAILABLE

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

   ! 0 !    A D D R E S S   O F   F I R S T   E N T R Y    !

   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

   ! 1   0  X! SECURITY CODE  ! NUMBER OF PENDING REQS. !

   STATE 3:  CLASS ALLOCATED, NO ONE WAITING ON CLASS
             NUMBER OF PENDING REQUESTS COUNTER MAY BE 0-255

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

   ! 1   1  X! SECURITY CODE  ! NUMBER OF PENDING REQS. !

   STATE 4:  CLASS ALLOCATED, SOMEONE WAITING (SUSPENDED)
             NUMBER OF PENDING REQUESTS COUNTER MAY BE 0-255

 ACTIONS TO BE TAKEN WHEN HANDLING A CLASS I/O OR GET REQUEST
 DEPEND ON THE CURRENT STATE OF THE CLASS QUEUE HEAD
 GET REQUESTS:
  STATE 1.  ABORT THE PROGRAM I000, NO CLASS.
  STATE 2.  RETURN THE DATA FROM CLASS BUFFER
  STATE 3.  SET THE SOMEONE WAITING BIT(BIT14), SUSPEND PROGRAM
  STATE 4.  ABORT THE PROGRAM I000, ONLY ONE PROGRAM MAY BE
            SUSPENDED PER CLASS.
 CLASS I/O REQUESTS:
  STATE 1.  STATE 3 IS SET UP, SECURITY CODE IS LOW 5 BITS OF
            PROGRAM ID NUMBER, COUNTER IS SET TO 1.
  STATE 2.  THE COUNTER AT END OF QUEUE IS INCREMENTED BY 1
  STATE 3.  THE COUNTER IS INCREMENTED BY 1.
  STATE 4.  THE COUNTER IS INCREMENTED BY 1.
 ON COMPLETION OF CLASS I/O REQUESTS:
  STATE 1.  ILLEGAL--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 AND THE SOMEONE WAITING BIT(BIT14)
            IS CLEARED.
```

CLASS I/O (COMPLETION)

The actual I/O part of Class I/O is the same as standard I/O. The differences show up again at completion. At this time, RTIOC extracts the class number from the request and uses it to find the class head. The class list is then searched to find its end and the newly completed request is linked into the queue, and the number of pending requests is decremented by one. The transmission log and device status are saved in the queue element head. At this time the buffer portion of a write request is returned to the system available

memory (SAM) pool. The completion processing is finished by checking bit 14 of the class word to see if some program has made a GET request on the Class. If bit 14 is set, the $SCD3 routine is called to reschedule the waiting program.

CLASS I/O (GET)

The class get request is passed to RTIOC at the entry point $GTIO. RTIOC checks that the buffer is totally within legal address space (no wrap around, etc.) and that the class number is allocated and has the proper security code. At this point, a test is made to see if there is a request to be gotten. If so, the minimum of the request buffer and the number of words in the class queue element is moved to the users buffer.

If the user has not requested that the class queue element be saved, it is released to SAM. The user is passed the transmission log and device status and sent on his way.

All this is fine but what if:

1. There is no queue element available at this time?

2. The bit is set indicating another program is already waiting?

If there is no pending queue element, the caller is suspended in the status 3 list with a pointer to the class, and bit 14 of the class word is set to show this.

If bit 14 is already set, it could be that the program went away (was aborted or suspended) without clearing the bit (the bit is only used to cut down overhead anyway), so an attempt is made to reschedule the program which caused the bit to be set ($SCD3). If this attempt is successful the current caller is aborted (IO10). The other option the user has is to set the no-wait bit on the GET call. In this case, he will get back either the ones complement of the number of requests pending or the first queue element if there is one. The ones complement is used because this makes zero into a −1, which is needed since zero is a legal number of pending requests and also a legal status and we want the user to be able to separate the two conditions (both are returned in the A-register). This concludes our discussion of Class I/O.

RESOURCE NUMBERS

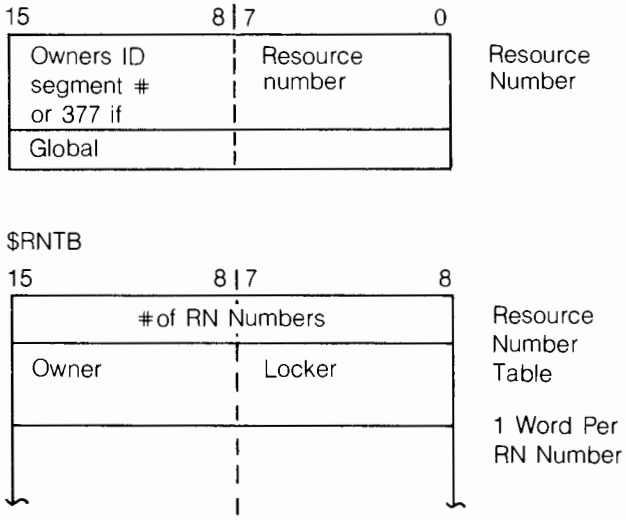
The resource number utility is also discussed in the referenced HP Journal article. Internally the RNRQ request is straightforward. The various resource number formats are given in Figure 2. When an allocate request is made, a search is done of the RN table (entry point $RNTB) from the high end down. (The reason for this will be given later.) When an unallocated number is found the new ownership information is recorded in the table, the table offset (Resource number) and owner ID is passed back to the caller. If the allocate bit was not set, or if it is, in any case, the next test is if the deallocate bit is set.

If the allocate bit is not set, an additional test is made to see if the RN passed in agrees with the table entry for that RN. If this test fails either RN02 (number illegal) or RN03 (owner is wrong) is generated.

If the deallocate is to be done, a further check is made to ensure that the caller is the owner (RN03) or that the ownership is global. If this test passes, the RN's entry in the table is cleared and any programs waiting for an RN number or for the particular RN are scheduled ($SCD3). Programs which are waiting for an RN number are suspended in the general wait list with a pointer to the head of the RN table. Programs which are waiting for a lock are suspended with a pointer to the entry in the RN table they are attempting to lock.

If the deallocate option bit is not set, a check is made to see if one of the lock bits is set. If so then, if the RN is not locked or is locked to the caller, then the requested lock flag (global or local) is set. If the RN is already locked, then either the program is suspended in the 3 list with a pointer to the RN or, if the no wait bit is set, control is passed back to the caller with the proper status.

Figure 2.



Owner/Locker = ID-segment number of program or 377 if global or 0 if not owned/locked

If the lock is successful or a lock is not coded, then the unlock or clear bit is checked in the option word. If the unlock option bit is set and the RN is either unlocked, globally locked or locked to the caller, then the low half of the RN word in the RN table is cleared and any waiters for the now clear RN are scheduled ($SCD3). If the RN is locked to some other user, a RN03 is generated and the program is aborted. This concludes the RNRQ request and our discussion of it.

LURQ

The LURQ request uses the RN allocate routine to allocate a RN from the low end of the RN table. Remember the RNRQ routine allocated from the high end of the table. The reason for this is that there are only 5-bits available for the RN number in the device reference table entry. These five bits define an RN which is further specified in the RN table. RN numbers allocated for LU booking are entered in the RN table as locked and allocated to the caller.

The LURQ request is handled as follows:

First, the device reference table is scanned and all locked LU's are checked to see if the caller is the locker. If so an RN has already been allocated and the call must be without wait (LU01). This test prevents a common form of dead lock where two programs lock the same two devices in different order and with two different calls. After this test is passed, each of the requested LU's is checked for existence (LU02) and availability. An LU is available if it is either not locked or is locked to the caller. If one is not available, the caller is suspended in the 3 list with a pointer to the RN table entry that the unavailable LU is locked with. At this point, all the LU's are available so, if an RN was not already allocated for the program, one is allocated now. If none is available, the program goes into the 3 list to wait for an RN (pointer to head of RN table). Once the RN is available, its number is put in each requested LU's entry in the DRT.

The LURQ request to unlock an LU checks for proper ownership of the lock (LU03) and if right clears the RN, then the RN is deallocated. Whenever an LU is unlocked, waiting programs are scheduled ($SCD3). Likewise when the RN is deallocated, programs waiting for an RN are rescheduled ($SCD3).

ABORT-TERMINATE CLEANUP

Whenever a program is aborted or terminates, the dispatcher calls the reentrant cleanup routine which in turn calls $TRRN. $TRRN releases all RN locks the terminating program has and likewise clears any LU locks. It then deallocates any RN numbers the program holds. After each of these operations, any waiting programs are rescheduled.

LU SWITCH

The LU switch table is a n-word table containing one word per switched LU. The first word is the tables length. The low 8-bits of each entry is the LU# of the LU to be switched and the high 8-bits are the LU# of the LU it is to be switched to. The switch table is used by LURQ and RTIOC whenever a program makes an LURQ or an I/O request and his Batch flag is set (word 21, bit 15 of his ID-segment-see Appendix A of your RTE manual). The LU switch table is set up by the LU command in the FMGR program when it is running a batch program. It is cleared by the EOJ command in FMGR.

This concludes this issues KNOW YOUR RTE. Please, help! What do you want to see in this column in the future? Do you want to see this column in the future? Only your cards and letters will keep it here.

SMUT

AN RTE SYSTEMS MAINTENANCE UTILITY

Larry W. Smith/DSD

The idea of an on-line interrogation routine for purposes of system design and debugging is not a new one but it's justification and usefulness in todays software systems is clearly becoming more and more necessary. This article describes the purpose and operation of a routine called "SMUT". The level of this article is tailored toward people with direct systems responsibility due to the nature and potential dangers (as it's name suggests) of this routine. This article can also be of benefit to others interested or beginning in systems work. It is hoped that this article will give you a flexible and convenient means to learn and evaluate Hewlett-Packard's Real-Time Systems in an effort to appreciate its capabilities and efficiency to solve your problems. It will also give you an insight into the overall design of RTE and it's ability to allow the user to design such things as 'SMUT'.

THE PURPOSE OF SMUT

Any real-time operating system is complex and difficult to learn the first time around. SMUT was created and designed not only to expedite and assist in learning RTE but also to do the following:

- Offer a sophisticated tool for use in RTE training classes to those involved in systems work.
- Gain complete control of an RTE system and interrogate systems activity.
- Help in design of your system.
- Help in debugging your system (drivers, data base, etc.).
- Exhibit the flexibility and control over RTE the user can obtain with sufficient experience.
- Untangle the user from difficult situations.

SMUT is not intended to take the place of system debug but merely to aid and supplement it. It's command structure is flexible and symbolic enough to permit the user to interrogate RTE at his/her own level.

SMUT is currently a 13K non-segmented program which does not use the Formatter and can run in any disc-resident partition of an RTE-II or III system or in an RTE-M system. It's length can be trimmed to 6K with the loss of convenience to certain disc commands and the absence of FMP SMUT command files. The main program (SMUT) is written in FORTRAN IV for your convenience with several small ASSEMBLY routines to more effectively accomplish certain operations. All in all, 13K of SMUT buys you the means to completely interrogate your RTE system with over 42 different command constructs. It comes complete with a 31 page users guide, a help processor, and sources and is orderable through LOCUS (Library Of Contributed User Software), paper tape (22682-18930, $50), cassette (22682-13330, $70).

THE CAPABILITIES OF SMUT

Since SMUT is primarily a software systems applications tool, not even a subset of its total capabilities could be described in this article. Thus, in this article we will describe it's general properties accompanied with illustrations of some of it's less sophisticated capabilities under the section titled "Examples of Effective SMUT Utilization".

Here's what SMUT offers you:

- Calculator to add, subtract, etc., in decimal, octal, and/or ASCII.
- Display, alter, scan, replace, copy, sum, and search blocks of memory.
- Read, write, scan, replace, copy, sum, initialize, and search disc (7900/7905/7920/flexible).
- List contents of system tables such as EQT, LUN, TO, INT, TAT.
- Entire set of HP-IB debugging commands.
- SMUT command files through FMP.
- Perform a disc modification relative to memory.
- Syntax help on an individual command basis (or all commands).
- Schedule other programs with wait (such as the editor).
- Output a character string to a device.
- Miscellaneous commands such as extended FMP calls, device control, and ASCII display modes.
- Command List and Log Device assignments.

All commands use REIO (re-entrant I/O interface) with a buffer size of 33 words and can be inputted from any device capable of input, including FMP type 0 files. After the com-

mand is entered, it is then torn apart by the system parse interface (PARSE) to determine the type and value of each parameter. If the command is not a SMUT command, it is sent directly to the system message processor interface (MESSS) as an actual RTE operator command. Thus, SMUT could be thought of as being equivalent to the system prompt (*) or any MTM terminal prompt (NN>). The initial input of all commands in the current version requires privileged processing since PARSE is always called and a few commands force privileged processing. All commands which initiate list output use the binary-to-ASCII interface routines 'CNUMD' and 'CNUMO' which also require privileged processing. If a zero-length record is sensed upon command input, only the previous SMUT command is Re-executed with the old parameters. If this occurs, a call to PARSE is bypassed and the appropriate computed branch is taken to execute the command.

HELP FOR SMUT

SMUT comes with a separate program called 'SHELP' (also written in FORTRAN IV) which gives either an alphabetical list of short command explanations or syntax help on a specific command. When help on a specific command is required (??,XX where XX is a command name), SMUT schedules SHELP in the queue with wait which then prints the appropriate help message on the list device.

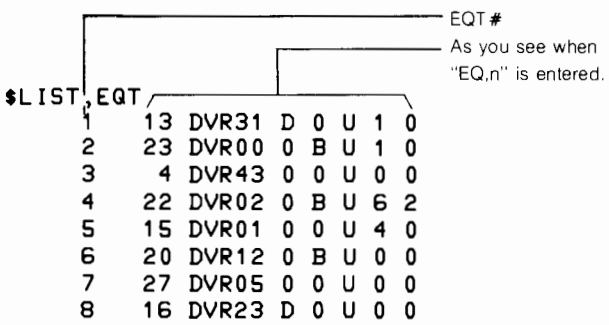
EXAMPLES OF EFFECTIVE SMUT UTILIZATION

The following examples will give a brief introduction on how to use some of the commands. If you purchase the routine, I would appreciate your comments and suggestions on its effectiveness to your application.

In all the below examples, operator inputs are shaded. The default prompt for SMUT is "$".

EXAMPLE 1: List all device EQT's, LU's, and TO's.

The following portions of the table list command 'LI' was implemented for system documentation purposes. The output device to which this listing will occur should be unbuffered.




```
$LIST,EQT
    1   13 DVR31 D 0 U 1 0
    2   23 DVR00 0 B U 1 0
    3    4 DVR43 0 0 U 0 0
    4   22 DVR02 0 B U 6 2
    5   15 DVR01 0 0 U 4 0
    6   20 DVR12 0 B U 0 0
    7   27 DVR05 0 0 U 0 0
    8   16 DVR23 D 0 U 0 0
    9   24 DVR00 0 0 U 1 0
   10   25 DVR00 0 0 U 1 0
   11   26 DVR00 0 0 U 0 0
   12   21 DVR12 0 B U 0 0
   13   72 DVR43 0 0 U 0 0
   14   73 DVR43 0 0 U 0 0
   15   74 DVR43 0 0 U 0 0
   16   75 DVR43 0 0 U 0 0
   17   76 DVR43 0 0 U 0 0
   18   77 DVR43 0 0 U 0 0
$LIST,LUN
 LU # 1 = E 2 S 1
 LU # 2 = E 1
 LU # 3 = E 0
 LU # 4 = E 4 S 6
 LU # 5 = E 5 S 4
 LU # 6 = E 6
 LU # 7 = E 7
 LU # 8 = E 8
 LU # 9 = E 9 S 1
 LU #10 = E10 S 1
 LU #11 = E 0
 LU #12 = E 7 S 1
 LU #13 = E 7 S 2
 LU #14 = E 0
 LU #15 = E 1 S 1
 LU #16 = E12
 LU #17 = E 0
 LU #18 = E 0
 LU #19 = E 0
 LU #20 = E13
 LU #21 = E14
 LU #22 = E15
 LU #23 = E16
 LU #24 = E17
 LU #25 = E18
 LU #26 = E 3
$LIST,TO
 TO#  1= 1000
 TO#  2=32767
 TO#  3=    0
 TO#  4=  100
 TO#  5= 5000
 TO#  6= 5000
 TO#  7=    0
 TO#  8= 5000
 TO#  9=    0
 TO# 10=    0
 TO# 11=    0
 TO# 12= 5000
 TO# 13=    0
 TO# 14=    0
 TO# 15=    0
 TO# 16=    0
 TO# 17=    0
 TO# 18=    0
```

It might be noted that some revisions of RTE will result in "OP CODE ERR" since $$CMD does not properly communicate to $MESS.

EXAMPLE 2: List system track ownership and partition status.

These commands list the symbolic contents of the track assignment table for LU 2 and 3 and partition status. This can be extremely valuable when the work area becomes cluttered and the Utilities (LOADR, FTN4, EDITR) cannot find available work tracks.

```
$LG,5
$LI,TAT
    0  SYSTEM
   29  EDITR
   30  SYSTEM
   35  EDITR
   39  EDI09
   40  EDITR
   41  EDI09
   42    --
   95    *LG0*
  100  FMGR
  202  D.RTR
$LI,ST
1    $$CMD
2    WHZAT
3    0
4    0
5    SMUT
6    0
7    0
8    0
9    0
```

Track # (Relative to 0)
Owner
Available Tracks
Literal indicating LG tracks
Partition #
Program in partition

EXAMPLE 3: Change terminal time-out less then 500 MS.

RTE-II/III will not allow a device time-out change on a terminal under 500 milliseconds. Thus, by changing word 14 of the base EQT to a negative value, a lower device time-out can be set.

```
$LIST,EQT,7
    1   34023      0       0
    2   34024  22721    9681  %Q
    3   34025  23011    9737  &
    4   34026  10027    4119
    5   34027   2402    1282
    6   34030    401     257
    7   34031  50674   20924  Q<
    8   34032 177657      81   /
    9   34033 121602   23678
   10   34034 121711   23607   I
   11   34035  24534   10588  )\
   12   34036    401     257
   13   34037  34307   14535  8G
   14   34040      0       0
   15   34041      0       0
$SM,34040B,-300
```

Word #
Memory Address
Octal
Decimal
ASCII

EXAMPLE 4: List system software entry points and documentation.

After your system has been generated, you might wish to obtain a list of entry points along with any extended NAM record documentation. This becomes particularly valuable to ensure you have the latest revision of the software.

```
$LI,ENT,$CIC
 $CIC  MRL   5715
$LI,ENT,EXEC
 EXEC  MRL  12436
$LI,ENT,..MAP,NAM
 ..MAP MIC 105222
$LI,ENT,FLOAT,NAM
 FLOAT MIC 105120
$LI,ENT,REIO,NAM
 REIO  DRL     T=17  S=33
 92001-16005 741120
$LI,ENT,$POWR
 $POWR MRL  20666
$LI,ENT,FAKE
 FAKE  ??
```

OCTAL Memory Address
Extended NAM record

Legend

MRL	—	Memory Resident Library
DRL	—	Disc Resident Library
MIC	—	Firmware Substitution
ABS	—	Absolute Value
T=nn		Track and Sector where
S=mm		Start of Relocatable Module Resides
??		Undefined entry point

EXAMPLE 5: Display EQT device list pointers.

The first word in each device EQT is the device suspended list pointer. The following example illustrates how to display all pointers repeatedly with only one SMUT command.

```
$DM,1651B
 1651      22      18
$CA,18,T,15
  416     270       0       0
$DM,1650B
 1650   33671   14265   79
$BM,33671B
$DM,,254,15
33671       0       0
33710       0       0
33727       0       0
33746   51762   21490   SR
33765       0       0
34004       0       0
```

Quotient in OCTAL & DECIMAL
Remainder in OCTAL & DECIMAL
(Calculate: 18*15)
(Set Base address for DM command)
Octal
Decimal
ASCII
All output formats of Memory and word display Commands have this format.
Memory Address (Relative to Base)

```
34023       0       0
34042       0       0
34061       0       0
34100       0       0
34117       0       0
34136       0       0
34155       0       0
34174       0       0
34213       0       0
34232       0       0
34251       0       0
34270       0       0
```

To look at this list repeatedly, simply enter any character which causes the device driver to return a zero-length record on input (such as 'CTRL-D', 'CR', or terminal time-out) and the previous SMUT command will be repeated.

EXAMPLE 6: List program ID segments.

Some elements of a programs ID segment cannot be changed on-line with standard operator commands. This can be done easily with the list ID command and the set memory command (SM).

```
                  WORD #
$LI,ID,SMUT      Memory Address
    1   41126         0       0
    2   41127        31      25
    3   41130    151670   11336   8
    4   41131    177657      81   /
    5   41132         0       0
    6   41133         0       0
    7   41134        50      40   (
    8   41135     46002   19458   L
    9   41136     71227   29335   R
   10   41137     71221   29329   R
   11   41140     71237   29343   R
   12   41141    100000   32767
   13   41142     51515   21325   SM
   14   41143     52524   21844   UT
   15   41144     20043    8227   #
   16   41145         1       1
   17   41146         0       0
   18   41147         0       0
   19   41150     25000   10752   *
   20   41151    177574     132   \
   21   41152     40020   16400   @
   22   41153     30004   12292   0
   23   41154     46000   19456   L
   24   41155     75326   31446   ZV
   25   41156         2       2
   26   41157       460     304   0
   27   41160     17000    7680
   28   41161         0       0
```

Long-ID Segment

```
$LI,ID,PRMPT
    1   37410   37440   16160   ?
    2   37411   37440   16160   ?
    3   37412   45310   19144   JH
    4   37413       2       2
    5   37414       0       0
    6   37415       0       0
    7   37416      12      10
    8   37417   45213   19083   J
    9   37420   45276   19134   J>
   10   37421       0       0
   11   37422       2       2
   12   37423  100000   32767
   13   37424   50122   20562   PR
   14   37425   46520   19792   MP
   15   37426   52041   21537   T!
   16   37427       0       0
   17   37430       0       0
   18   37431       0       0
   19   37432   25000   10752   *
   20   37433  177574     132    \
   21   37434       0       0
   22   37435     200     128
   23   37436       0       0
   24   37437       0       0
$LI,ID,FMGR7
    1   42246   52024   21524   T
    2   42247   43115   17997   FM
    3   42250   43522   18258   GR
    4   42251   33425   14101   7
    5   42252   52024   21524   T
    6   42253   57145   24165   ^E
    7   42254      30      24
    8   42255      72      58    :
    9   42256    3602    1922
$LI,ID,PROGX
 NO SUCH PROG
```

Memory Resident ID Segment

Short-ID Segment

EXAMPLE 7: Un-purging an FMP file.

After an FMP file has been purged, FMP replaces the first two characters of the file name in the directory with (−1). Thus, if the directory can be located and changed back to the original entry and the disc has not been packed, the file can be recovered.

```
*ON,FMGR
:PU,WELCOM:LS:-2
:EX
$END FMGR
*ON,SMUT
$LI,TAT
```

```
  0  SYSTEM
 29  EDT07
 31    --
 32  SYSTEM
 41    --
 42  SYSTEM
 45    --
 51  SYSTEM
 55    --
 66   *LGD*
 75  FMGR
202  D.RTR
```

List TAT to find Directory Track (D.RTR=202)

```
$RD,202,0,2,6144 — (Read track into work area)
$FW,LC,1,6144,ALL— (Find location of purged directory entry)
1810                    Assumed (-1) indicating a purged
$DW,1809,1824           file to be eliminated on next
1809  177777      1     pack or Store Operation
1810   46103  19523  LC
1811   47515  20301  OM
1812       3      3   #
1813     115     77
1814      40     32
1815       6      6   &
1816       0      0
1817   46123  19539  LS
1818       0      0
1819       0      0
1820       0      0
1821       0      0
1822       0      0
1823       0      0
1824       0      0
$SW,1809,WE ———— (Restore to original entry)
$WD,202,0,2,6144 — (Post directory back on disc)
```

This type of function must be done with absolutely no FMP activity.

EXAMPLE 8: Scheduling a program immediate with wait.

Since SMUT is designed as an on-line maintenance and debug tool, it becomes handy to be able to schedule other programs (such as FMP) and return to SMUT after the work has been completed.

```
$SC,EDITR —— (Terminal LU defaults to SMUT's log-on lu)
SOURCE FILE?
/&LUGET:LS:-15
ASMB,R,L
/
     :
     :
/ER&LUGET
END OF EDIT
$ ———————— (BACK IN SMUT)
```

EXAMPLE 9: Copying disc to tape for back-up.

SMUT has the capability to dump any device to any device. This example shows how to copy disc to mag tape.

```
$CN,8,4 ———— (Rewind MT)
$IB,8 ———— (Specify output LU)
$IO,RTE-3/64K FFP 7905 DATA ACQ SYSTEM
$CD,2,,202,8 \                        \
$CN,8,1       (Copy from disc         Write a header
$CN,8,4       to MT)                  to tape
               (Write EOF)
               (Rewind MT)
```

EXAMPLE 10: List HP-IB Bus Configuration.

This command lists any HP-IB configuration for any number of IBI cards. It lists each devices auto-addressable talk and/or listen addresses.

```
$LIST,HPIB
                          Device Address
 LUN   CHAN   SUBCHN    TALK      LISTEN
  13     13       3   67   C    35   #
  17     13       5   69   E    37   %
  18     13      12   74   J    42   *
  19     13      17   79   O    47   /
  20     13      20   80   P    48   0
  21     13  DIRECT   64   @    32
  22     14       7   71   G    39   '
  23     14      15   77   M    45   -
  24     14      31   89   Y    57   9
  25     14      21   81   Q    49   1
  26     14  DIRECT   64   @    32
         \
          OCTAL I/O Channel
```

EXAMPLE 11: Transfer to a SMUT Command File.

SMUT has the capability to execute any command from an FMP file as follows:

```
$TR,COMND,SC,-15 — CR
                 \— Security
```

This can be particularly useful for pre-defining a series of commands such as to exercise an instrument on the BUS and executing them with one command.

CONCLUSION

The effectiveness of this routine really depends on you. Thus, we would appreciate any useful applications in which you have used SMUT be sent to:

Larry W. Smith
Technical Center

If your inputs are not duplicates, I will include them in the next revision of the Users Guide.

HOW TO RECOVER YOUR EDITED SOURCES AFTER A SYSTEM REBOOT

Al Liu/DSD

Have you ever felt your whole day's work going down the drain when you find it is necessary to reboot the system while you are in the midst of an extensive editing job? Your updates or new entries have not yet been written to your FMGR file. They are still on the logical source tracks.

The following transfer file can help you to recover your sources after you have rebooted the system.

```
: SE,1
: LS,2,1G
: RU,EDITR
: CA,1,1G,+,1
: IF,,EQ,,-4
::
```

The transfer file sets up the starting track # for EDITR to scan in the logical source (LS) track area. Turn on the transfer file from FMGR by entering ::namr where namr is the FMGR file name of the transfer file.

When EDITR asks "SOURCE FILE ?", enter a space and carriage-return. This causes EDITR to scan and display the first record in the starting logical source track. If the display(s) from EDITR are those of your edited file, enter "EC namr" or "ER" to recover these sources into a FMGR file.

If the display(s) from EDITR are not of your editing, enter "A" to abort EDITR for that time. The transfer file will repeat at the LS command with the starting LS track # incremented by 1. This repeating process enables you to scan the LS tracks one by one until you have reached your edited file.

After you have recovered your editing, remember to terminate the infinite looping of the transfer file by aborting FMGR immediately (OF, FMGR, 1).

NEW CONTRIBUTED PROGRAMS

Melanie Van Vliet/DSD

Vacation time is about over and we expect a bit more action for our next column. Remember all things great and small are welcomed additions to the LOCUS. The hours you invested in programming can be shared by many through the contributions you make, in turn you may save hours of programming by implementing one of the available programs.

All first time entries are awarded the traditional Wall Plaque engraved with the contributor's name and a program name disc for each accepted program. A complimentary program of the contributors choice is also included with each and every accepted contribution to the LOCUS.

At the back of this issue of the COMMUNICATOR 1000 you will find a complete set of Contributor documentation forms, and guidelines for submitting a program. We hope to hear from you.

This article serves as an update for the Data Systems LOCUS Program Catalog (22000-90099).

The new contributed programs listed below are now available. Contact your local HP Sales Office to order Contributed Library material, or (if you are in the U.S.) you can use the Direct Mail Order form at the back of the COMMUNICATOR 1000.

22682-18969 TELL & TELL ALL INTER-TERMINAL MESSAGE SENDING

The SYSTK program enables communication between two terminals in a MTM environment. Up to 10/80 character message lines may be sent from one terminal to another. Control may be maintained at one terminal or passed along with / or without a message to the second terminal. Functions are determined through 4 commands preceeded by a program prompt (?).

The SMESS program enables system console (or initiator) to output a message to all TTY's in a MTM environment. Up to 10/80 character message lines may be sent / text-send command. The first message sent also includes the Gregorian date and current time, routines TYME and JLIAN. SMESS functions are determined through 3 commands preceeded by a program prompt (?). Standard RTE-II/III configuration with %MTM (optional).

22682-18969	PT	$10.00
22682-13369	Cass	$35.00

The following program has been reinstated in the LOCUS Program Catalog:

22682-*38 DOS-IIIB INTERACTIVE EDITOR**

SOFTWARE SAMANTHA
Messrs. Software Samantha
Care of Communicator
1000 (9600) Group
HP Data Systems Division

Dear Sirs:

Re: Algol Compiler RTE-II/DOS # 24129-60001 Rev. C

Because of unsatisfactory results obtained from the contributed Chess Programme #22660, using the above compiler in a RTE-II environment, I discovered a number of errors in the Algol Compiler.

First the procedure Bakup 2 in the minimax game tree is reproduced here in simplified form.

The result from "DTEST" shows that 1500 < −30000. This is because the generated compiler sequence causes an integer overflow condition.

In "CTEST", which has the same programme logic and resembles "BAKUP 2" closer, another error shows-up. The first if-condition T < AB(2-1) is also tested for the "Equal" condition by means of a SZA, while the second if-condition T <= AB(2-2) is not tested for the equal sign.

Second the procedure PROMO(TB,PTR) is reproduced in programme "ALG 5". Instead of the I-Loop executed once, it becomes endless.

Another problem related to the Algol Compiler is as follows:−
A thermal line printer type 9866B with micro-circuit interface is used with driver DVR12. During Algol compilation and listing the line printer control character is not skipped and the resultant listing skips lines and the first line characters are cutt-off. The assembler and Fortran IV listings are correct.

Not connected with the Algol Compiler is another problem. Using the micro-circuit interface and driver DVR00 or DVR12, the back arrow " ◀ " or the RTE − Basic semicolon is not recognised and the line feed can not be suppressed. This problem does not excist using the line printer with a 8 bit duplex punch interface.

The 9600A system (WO # 22160890 − System Serial # 1640A70069) was supplied with a 98668 line printer and micro-circuit interface.

Since I like to make use of the plotting capability of the 9866B printer and no programming information has been received, I would appreciate some information about this.

Since we as a group frequently use Algol programming and we are now reluctant to use it, we hope that you are able to solve this matter to our satisfaction and are awaiting an early reply.

Yours faithfully,
D.A. Van Den Eijkel
Sen. Officer (Special Prodjects)
Protection, Telecommunications & Control
Elecricity Suppply Commission
P.O. Box 103
Transvaal
Rep. of South Africa

Dear Mr. D.A. Van Den Eijkel,

Program DTEST, below, indeed does return this result 15000 <= −30000. This is due to the integer overflow resulting from the test for T<= −30000 when T = 15000. The condition results when 30000 is added to 15000 for the comparison to zero. The philosophy here, and that which is commonly followed by compilers is the assumption that the user will take into consideration the possibility of overflow when programming. That is, overflow checks are not made following each arithmetic operation.

```
HPAL,L,A,''DTEST''
BEGIN INTEGER T:=15000;
IF T<30000 THEN
   BEGIN IF T<=-30000 THEN
    WRITE(1,#("T<-30000))
     ELSE WRITE(1,#("T>-30000))
END$ OF DTEST
```

Due to the method used in Algol to parse the command string the generated code does not always directly reflect the Algol statement, as program CTEST illustrates. Here an equivalent Boolean expression is generated rather than the actual expression in the IF statement.

```
ALGOL STATEMENT        CONDITION TESTED
IF T < AB(2-1)         NOT[ T >= AB(2-1) ]
IF T <= AB(2-2)        NOT[ T <  AB(2-2) ]
```

This is due to the method used to parse the command string in Algol.

```
HPAL,L,A,"CTEST"
BEGIN INTEGER ARRAY AB[0:3];
INTEGER T,A;
AB[0]_30000;
AB[1]_-30000;
T_15000;
IF T<AB[2-1] THEN
BEGIN IF T<=AB[2-2] THEN
BEGIN A_1;    WRITE(1,#("A=",I2),A) END
ELSE BEGIN A_0;      WRITE(1,#("A=",I2),A) END
END END$
```

Next consider the program ALG5, below. Algol users please note that in Algol if the increment or final value in a DO LOOP is a simple variable, changes to the variable will affect the increment or final value. In ALG5 the final value of the loop, PTR, is being incremented by one each time through the loop, creating an infinite loop.

```
HPAL,L,A,"ALG5"
BEGIN INTEGER I,J,TB,PTR;
      INTEGER ARRAY TREE[1:10,1:12];
  TB_5;       PTR_6;
   FOR I_TB+1 TO PTR DO
    BEGIN TREE[I,2]_TREE[I,2] OR 4;
     FOR J_1 TO 3 DO
       BEGIN PTR_PTR+1;     TREE[PTR,2]_TREE[I,2];
       CASE J BEGIN
        TREE[PTR,1]_TREE[I,1] OR %40000;
        TREE[PTR,1]_TREE[I,1] OR %100000;
        TREE[PTR,1]_TREE[I,1] OR %140000;

  END END END
END$ OF ALG5
```

Next, the problem you describe with the compiler not outputting a carriage control character for the line printer has been corrected in revision D of the Algol compiler. The problem is documented in SST report 3596.

Not connected with the Algol compiler, the recognition of the back arrow and Basic semicolon characters is software dependent only, that is, determined by the driver and not the interface card. DVR00 suppresses the carriage return and line feed if the back arrow is the last character in the buffer, whereas DVR12 treats it as an underline character which is printed by the 9866.

The 9866B thermal printer has a plot mode which allows one to use the printer as an incremental plotter. Plotting is accomplished by printing a specified dot pattern in a specified column for any one point on the plot. As of yet Hewlett-Packard does not provide a software plotting package for the 9866B, user would need to write the software to utilize the plotting capabilities. The information on programming in plot mode can be found in the *Hewlett-Packard 9866A/B Printer Peripheral Manual,* part number 9866-90001.

PATH FROM SEGMENT TO MAIN IN FORTRAN

I would also like to mention an easy method to transfer control from a segment back to the main in FORTRAN IV. Consider the programs "MAIN" and "SEG1" below. A label variable LABEL is declared and defined in the main program, then passed to the segment through COMMON (note that it could be passed as a parameter in the EXEC 8 call to load the segment). Now the segment may return control back to the main starting at the address defined by LABEL with simply a GO TO LABEL statement.

```
FTN4,L
      PROGRAM MAIN
      INTEGER SEG1(3),IPARM(5)
      COMMON ILU,LABLE
      DATA SEG1/2HSE,2HG1,2H   /
      CALL RMPAR(IPARM)
      ILU=IPARM(1)
      ASSIGN 100 TO LABLE
      CALL EXEC(8,SEG1)
100   WRITE(ILU,1000)
1000  FORMAT(" BACK TO 100 IN MAIN!!!!")
      END

FTN4,L
      PROGRAM SEG1,5
      COMMON ILU,LABLE
      WRITE(ILU,1000)
1000  FORMAT(" INTO SEG1!!!!")
      GO TO LABLE
      END
```

SETTING TIME ON BOOT-UP

From **Jim Luscher**, Data Terminals Division, we have a program which prompts the operator to enter the current date and calculates the Julian date. The correct form of the TM command is presented to the operator and the program then hangs in a loop waiting until a date different from the default boot-up date is entered. The program can be automatically invoked at boot-up by inserting the following lines in the WELCOM file:

```
:RP,SETTM
:RU,SETTM
:OF,SETTM
```

Note the program must be saved as a type six file.

```
FTN4,L
      PROGRAM SETTM
C     * PROGRAM TO CALCULATE THE JULIAN DATE AND PROMPT
C     * THE OPERATOR TO ENTER THE "TM" COMMAND (SET TIME)
C     * UPON BOOTSTRAPPING OF THE RTE OPERATING SYSTEM.
      INTEGER A
      DIMENSION IN(3),A(12,4),ITIME(5)
      DATA A( 1,1)/1HJ/,A( 1,2)/1HA/,A( 1,3)/1HN/,A( 1,4)/31/
      DATA A( 2,1)/1HF/,A( 2,2)/1HE/,A( 2,3)/1HB/,A( 2,4)/29/
      DATA A( 3,1)/1HM/,A( 3,2)/1HA/,A( 3,3)/1HR/,A( 1,4)/31/
      DATA A( 4,1)/1HA/,A( 4,2)/1HP/,A( 4,3)/1HR/,A( 4,4)/30/
      DATA A( 5,1)/1HM/,A( 5,2)/1HA/,A( 5,3)/1HY/,A( 4,4)/31/
      DATA A( 6,1)/1HJ/,A( 6,2)/1HU/,A( 6,3)/1HN/,A( 4,4)/30/
      DATA A( 7,1)/1HJ/,A( 7,2)/1HU/,A( 7,3)/1HL/,A( 7,4)/31/
      DATA A( 8,1)/1HA/,A( 8,2)/1HU/,A( 8,3)/1HG/,A( 8,4)/31/
      DATA A( 9,1)/1HS/,A( 9,2)/1HE/,A( 9,3)/1HP/,A( 9,4)/30/
      DATA A(10,1)/1HO/,A(10,2)/1HC/,A(10,3)/1HT/,A(10,4)/31/
      DATA A(11,1)/1HN/,A(11,2)/1HO/,A(11,3)/1HV/,A(11,4)/30/
      DATA A(12,1)/1HD/,A(12,2)/1HE/,A(12,3)/1HC/,A(12,4)/31/
C     * GET CURRENT DATE (SYSTEM ASSEMBLY)
      CALL EXEC(11,ITIME,IYOLD)
      IDOLD=ITIME(5)
C     * GET MONTH FIRST
1     WRITE(1,10)
10    FORMAT(" ENTER MONTH (JAN/FEB/ ... /DEC)")
      READ(1,20) (IN(K),K=1,3)
20    FORMAT(3A1)
C     * LOOK FOR A MATCH IN TABLE "A"
C     ** LOOP THROUGH EACH MONTH
      DO 50 M=1,12
C     ** AND TEST ALL THREE CHARACTERS OF NAME
      DO 40 J=1,3
      IF (A(M,J).NE.IN(J)) GO TO 50
40    CONTINUE
C     * A MATCH !!
      GO TO 100
50    CONTINUE
C     * NO MATCH !!
      GO TO 1
C     * NOW GET DAY OF MONTH
100   WRITE(1,101)
101   FORMAT(" ENTER DAY OF MONTH (1-31)")
      READ(1,*)IDAY
      IF (IDAY.LT.1.OR.IDAY.GT.A(IMONTH,4)) GO TO 100
C     * GET YEAR
150   WRITE(1,151)
151   FORMAT(" ENTER YEAR (YYY)")
      READ(1,*) IYEAR
      IF (IYEAR.LT.1977) GO TO 150
C     * LEAP YEAR ??   (SHORTEN FEBRUARY IF NOT !)
      IF (MOD(IYEAR,4).NE.0) A(2,4)=28
C     * COUNT DAYS IN YEAR
      ISUM=0
C     * ADD ALL MONTHS UP TO THIS ONE
      IF (M.LT.2) GO TO 200
      M=M-1
      DO 180 K=1,M
180   ISUM=ISUM+A(K,4)
C     * ADD IN DAYS IN THIS MONTH
200   ISUM=ISUM+IDAY
      WRITE(1,300) IYEAR,ISUM
300   FORMAT(/" ENTER THE FOLLOWING COMMAND TO SET TODAY'S DATE:"
     X       /"     (INSERT CORRECT HOUR/MIN./SEC =>"
     X        "24 HOUR CLOCK!!)"/
     X        "=TM,",I4,",",I3,",",HOUR,MINUTE,SECOND"/)
C     * WAIT FOR OPERATOR RESPONSE ??
400   CALL EXEC(11,ITIME,IYR)
C     * DATE UPDATED YET ??
      IF (ITIME(5).EQ.IDOLD.AND.IYR.EQ.IYOLD) GO TO 400
C     * EXIT
      CALL EXEC(6)
      END
```

If you have any questions, suggestions, or comments about your HP 1000 (9600) system, please address them to:

SOFTWARE SAMANTHA
c/o Communicator 1000(9600) Group
HP Data Systems Division
11000 Wolfe Road
Cupertino, CA. 95014

THE MILLION BYTE 21MX

Bill Elmore/DSD

Hewlett-Packard's leadership in semiconductor memory brings a significant new capability for small computer customers: One million bytes of fault control memory in a high performance computer that occupies only 12-1/4 inches of rack space!

Two new products, both based on state-of-the-art 16K bit NMOS RAM's, make this possible: The HP 12747A 128K byte memory module, and the 2101C fault control memory system. These new products make very large memory systems a practical matter by significantly increasing reliability and packaging density.

BENEFITS

The two new products bring all the benefits of very large memory systems to 21MX users.

An important benefit is greatly improved throughput for multi-user and multi-programming applications. By implementing programs in main memory, disc swapping and other inherently slow techniques can be minimized. Remember that RTE-III can support up to 64 partitions of up to 64K bytes each.

The fault control capability provides a tenfold improvement in Mean Time Between Failure (MTBF) of 21MX memory. This means that large memory systems are now viable from a reliability standpoint. For example, a 2113 computer with one million bytes of fault control memory should exhibit a long term MTBF of 6,000 to 8,000 hours, or roughly one year of operation between failures. Fault control capability can be a key benefit to smaller memory users as well. Applications that place a high premium on reliability include distributed processing, data communications front ends, data acquisition front ends, and remote, unattended applications.

The cost of the new memory systems even makes replacement of fixed head or small moving head discs a practical thing to do. A 21MX with one million bytes is substantially faster, quieter, and more reliable than a disc system, and very close in price. The main memory also has a greater environmental range, consumes less power, and requires less maintenance than any kind of disc system.

Another key benefit of the one million byte 21MX is that it is entirely self-contained in a 12-1/4 inch mainframe with fourteen fully powered I/O slots, and requires no special wiring, raised floors, air conditioning, or space-consuming equipment bays. The wide environmental range (0 to 55°C operating temperature, 20 to 95% humidity at 40°C) enables the 21MX and its memory to go almost anywhere in order to get the job done.

128K BYTE MEMORY MODULE

The HP 12747A 128KB memory module is a standard performance module that is compatible with 21MX M,K, and E-Series computers and the 2102B standard performance controller or the 2101C fault control memory system controller. Housed on the same size PC board as the 32KB modules, the 12747A quadruples the memory capacity of 21MX systems.

The 128KB module is based on a new 16K bit N-channel MOS memory chip that has been fully qualified to HP's stringent reliability standards. The 16K RAM will undergo the same exhaustive part condition and testing processes as the 4K RAM's.

FAULT CONTROL MEMORY SYSTEM

The fault control memory system consists of the 2101C controller and the 12779A and 12780A check bit array boards. The system detects and corrects all single bit faults, detects all double-bit faults, and detects and reports most multiple bit (3 or more) faults.

The fault control system uses 5 extra bits per word to compute a hamming error detection and correction code, which functions like a very sophisticated parity system. There is a unique code associated with each data word. When a word in memory is read, a new code is generated for that word and compared to the code stored with the word. If a single bit fault is present, it is automatically corrected; all double bit faults are reported as parity errors.

(Many error correction systems don't guarantee detection of all double bit errors, since they only utilize 21 bits. The HP fault control system can do this because a 22 bit word is used.)

The 5 extra bits per word are physically housed in check bit array boards which mount in the memory section of the computer mainframe. The HP 12779A can support up to 256K bytes of memory; the 12780A, up to 512K bytes.

Fault control memory systems are configured with a 2102C memory controller, one or more check bit array boards (12779A or 12780A) and the appropriate number of standard performance memory modules (16KB 12998A, 32KB 13187B, or 128KB 12747A). For ordering convenience, several packages are available, consisting of Dynamic Mapping hardware and firmware, check bit array boards, and 128KB memory modules.

State-of-the-art 16K bit NMOS RAM's have made it possible for HP to design this advanced memory system.

With the HP 12747A 128K byte memory module, and the 2102C Fault Control Memory System, very large, reliable memory systems are a reality in 21MX M-Series and E-Series computers. Sound exciting? You bet!

DOCUMENTATION

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** 1000.

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** 1000.

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.

1000 SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02170-90006	HP 1000 Computer System Installation and Service	$ 2.50	7/77*R	
02172-90005	Getting Started with Your HP 1000 Disc Based Computer System (for A computers)	4.00	6/77	
02172-90010	Getting Started with Your HP 1000 Disc Based Computer System (for B computers)	2.50	8/77*N	
02173-90007	Getting Started with Your HP 1000 System: Models 20 and 21	2.50	8/77*N	
91780-93001	RJE/1000 Programming Manual	9.50	11/76	6/77

RTE SYSTEMS MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02313-93002	RTE 2313B Analog-Digital Interface Subsystem Operating and Service Manual	$30.00	8/76	
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 DVR 74 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	91200B TV Interface Kit; Programming and Operating Manual	4.50	7/75	1/76
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	
09601-93015	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	5/76	

RTE SYSTEMS MANUALS (Continued)

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
09610-93003	ISA FORTRAN Extension Package Reference Manual	4.50	7/76	
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	
12732-90001	RTE Driver DVR33 Programming Manual	2.00	2/77	
13197-90001	RTE Driver DVR36 Programming and Operating Manual	3.00	9/76	
24998-90001	DOS/RTE Relocatable Library Reference Manual	10.00	5/77	
25117-93003	RTE System Driver DVR24 for HP 7970 Series Digital Magnetic Tape Unit	1.00	8/74	
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 2313B Subsystem	2.50	8/74	
29028-95001	RTE HP 2610A/2614A Line Printer Driver	1.50	8/73	
29029-95001	Real-Time Executive System Driver DVR00 for Multiple Device System Control Small Programs Manual	1.50	11/75	
29100-93001	RTE System Driver DVR40 (29100-60041) for HP 12604B Data Source Interface Programming and Operating Manual	1.00	8/76	
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	12/76	5/77
59310-90063	DVR37 Manual	3.50	6/77	
59310-90064	HP-IB Interface Bus I/O Kit Users Guide	8.50	4/77	6/77
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	
91700-93001	Distributed System CCE Operating Manual	20.00	5/77*R	
91705-93001	Distributed System SCE/5 Operating Manual	15.00	12/76	
91200-90005	RTE Driver DVA13 for TV Interface (HP 91200B)	1.50	5/77	
92001-90015	RTE DVR05 for 264X Terminals	2.00	9/76	
92001-93001	RTE-II Software System Programming and Operating Manual	10.00	7/77	8/77
92060-90004	RTE-III Software System Programming and Operating Manual	12.00	7/77	8/77
92060-90005	RTE Assembler Reference Manual	7.00	12/76	
92060-90009	RTE-III General Information Manual	4.00	2/76	
92060-90010	RTE Batch/Spool Monitor and Operating System Pocket Guide	3.00	4/77	
92060-90012	RTE: A Guide for New Users	6.50	7/76	
92060-90013	Batch-Spool Monitor Reference Manual	9.50	3/77	
92060-90014	RTE Interactive Editor Reference Manual	6.00	5/77*R	
92060-90017	RTE Utility Programs	3.00	3/77	
92060-90020	RTE On-Line Generator	15.00	7/77	
92064-90002	RTE-M Programmer's Reference Manual	14.00	3/77	7/77
92064-90003	RTE-M System Generation Reference Manual	7.50	3/77	7/77
92064-90004	RTE-M Editor Reference Manual	6.00	1/77	3/77
92200-93001	RTE System Driver DVR12 for HP 2607A Line Printer Programming and Operating Manual	1.00	8/74	
92200-93005	Real-Time Executive Operating System Drivers and Device Subroutine Manual	5.00	3/77*R	
92202-93001	RTE System Driver DVR23 for HP 7970 Series Digital Mag Tape Units Programming and Operating Manual	1.00	8/74	
92400-93001	92400A Utility Library Subroutine for Sensor-Based Diagnostics	7.50	11/76	
93005-93005	Thermal Line Printer Subsystem for Driver DVR00 (RTE)	2.50	12/74	

HARDWARE MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02108-90002	HP 21MX Reference Manual	$ 5.50	6/76	7/76
02108-90006	HP 21MX Installation and Service Manual	10.00	7/76	
02108-90004	HP 21MX Operators Manual	5.00	7/76	
02108-90017	21MX M-Series Computer Engineering and Reference Documentation	125.00	5/77*R	
02108-90027	21MX K-Series Computer Engineering and Reference Documentation	100.00	5/77*R	
02109-90001	HP 21MX E-Series Computer Operating and Reference Manual	8.00	7/77*R	
02109-90002	HP 21MX E-Series Installation and Service Manual	15.00	8/76	3/77
02109-90006	HP 21MX M- and E-Series I/O Interfacing Guide	7.00	7/77*R	
12979-90007	HP 12979A I/O Extender Operating and Reference Manual	5.00	12/75	
12979-90006	HP 12979A I/O Extender Installation and Service Manual	15.00	6/77*R	
12990-90003	HP 12990A Memory Extender Installation and Service Manual	5.50	4/76	8/76

LANGUAGE MANUAL

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02100-90140	Decimal String Arithmetic Routines	$ 6.50	2/77	
02108-90032	HP 21MX M-Series Computer RTE Microprogramming Reference Manual	15.00	10/76	
02108-90034	HP 21MX M-Series Computer RTE Microprogramming Pocket Guide	2.75	1/77	
02109-90004	21MX E-Series RTE Microprogramming Reference Manual	20.00	3/77	
02109-90008	21MX E-Series Computer RTE Microprogramming Pocket Guide	2.50	11/76	
02116-9014	HP Assembler Manual	6.50	8/75	
02116-9015	HP FORTRAN Manual	6.00	1/77	
02116-9016	Symbolic Editor	4.50	2/74	
02116-9072	ALGOL Reference Manual	10.00	11/76	
12907-90010	Implementing the HP 2100 Fast FORTRAN Processor	1.00	7/76	
24307-90014	DOS-III Assembler Reference Manual	8.00	7/74	11/75
92060-90005	RTE Assembler Reference Manual	7.00	12/76	
92060-90016	Multi-User Real-Time BASIC Reference Manual	12.00	2/77	4/77
92060-90023	RTE FORTRAN IV Reference Manual	10.00	7/77	
92063-90001	IMAGE/1000 Data Base Management System Reference Manual	9.00	2/77	7/77*R
92065-90001	RTE-M Real-Time BASIC Language Reference Manual	8.50	2/77	7/77

SOFTWARE UPDATES

Following are cross-reference lists of the available 92001B, 92060B, 92062A, and 92064A (options 20 & 40) software modules, the media on which the software modules are distributed, and the date code or revision of each module up to, and including level 1726. Software modules updated since the last issue are indicated for easy reference.

NOTE:

For each module, interdependencies with other modules may exist (i.e., any updated module may require other updated modules to function properly).

SOFTWARE MODULE NUMBERS: 92001B LEVEL 1726 (RTE II)

The following modules are also available on a 7900 RTE Master Software Disc (#92001-13001), or a 7905 RTE Master Software Disc (#92001-13101).

PAPER TAPE	MODULE	DESCRIPTION	CARTRIDGE	DATE CODE
02607-16004	!S4L07	24K SIO LINE PRINTER DRIVER	92001-13305	1538
09601-16021	%DVR15	RTE 7261A DRIVER	92062-13304	A
12732-16001	%DVR33	FLEXIBLE DISC DRIVER	92062-13304	1726
12970-16004	!S4MT1	24K SIO MAG. TAPE DRIVER	92001-13305	1550
20747-60001	%DVR30	RTE FIXED HEAD DISC DRIVER	92062-13305	C
20808-60001	%CAL10	CAL. PLOTTER DRIVER	92062-13302	B
20810-60001	%CALIB	CAL. PLOTTER LIBRARY	92062-13302	C
20875-60001	%1FTN	FORTRAN MAIN CONTROL	92060-13308	E
20875-60002	%2FTN	FORTRAN PASS 1	92060-13308	E
20875-60003	%3FTN	FORTRAN PASS 2	92060-13308	E
20875-60004	%4FTN	FORTRAN PASS 3	92060-13308	F
20875-60005	%5FTN	FORTRAN PASS 4	92060-13308	F
24129-60001	%ALGOL	RTE/DOS ALGOL PART 1	92060-13305	1643
24129-60002	%ALGL1	RTE/DOS ALGOL PART 2	92060-13305	C
24153-60001	%FF.N	RTE/DOS FORMATTER	92060-13303	C
24306-60001	%DECAR	DOSM ST ARITH PK	92060-13303	A
24998-16001	%RLIB1	RTE/DOS LIBRARY PART 1	92060-13302	1726
24998-16001	%RLIB2	RTE/DOS LIBRARY PART 2	92060-13302	1726
24998-16002	%FF4.N	FORTRAN IV FORMATTER	92060-13303	1726
25117-60499	%DVR24	RTE 7970 7T MAG. TAPE DRIVER	92062-13305	D
29013-60001	%DVR31	RTE 7900A DISC DRIVER	92062-13305	1710
29028-60002	%DVR12	RTE 2767A DRIVER	92062-13303	A
29029-60001	%DVR00	RTE TTY/PUNCH/PHOTO READER	92062-13302	1642
29030-60001	%DVR11	RTE 2892A CARD READER DRIVER	92062-13303	1710
29100-60017	!S4LP	24K SIO LINE PRINTER	92001-13305	A
29100-60018	!S4SYD	24K SIO SYSTEM DUMP	92001-13305	A
29100-60019	!S4PHR	24K SIO PHOTO READER	92001-13305	A
29100-60020	!S4PUN	24K SIO TAPE PUNCH	92001-13305	A
29100-60022	!S4L67	24K SIO 2767 LINE PRINTER	92001-13305	A
29100-60023	!S4MT2	24K SIO 7970 MAG.TAPE	92001-13305	A
29100-60049	!S4MT3	24K SIO MAG. TAPE	92001-13305	A
29100-60050	!S4TER	24K SIO TERMINAL PRINTER	92001-13305	A
59310-16002	%1DV37	RTE HP-IB WITHOUT SRQ	92062-13304	1726
59310-16003	%2DV37	RTE HP-IB WITH SRQ	92062-13304	1726
59310-16004	%HPIB	HP-IB DEVICE SUBROUTINE	92062-13304	1710
59310-16005	%SRQ.P	SRQ.P TRAP UTILITY	92062-13304	1710
72008-60001	%1DV10	COMP. 7210A PLOTTER DRIVER	92062-13302	A
72009-60001	%2DV10	MIN. 7210A PLOTTER DRIVER	92062-13302	A
91200-16001	%DVA13	91200A DRIVER	92062-13303	1648
91200-16002	%TVLIB	91200A VIDEO MONITOR LIBRARY	92062-13303	1648
91200-16004	%TVVER	91200A TV INTERFACE VERIFIER	92062-13303	1648
92001-16002	%LDR2	RTE LOADER	92001-13301	1726
92001-16003	%MTM	MULT. TERMINAL MONITOR	92001-13301	H
92001-16004	%2DP43	POWER FAILURE DRIVER	92001-13301	1633
92001-16005	%SYLIB	RTE SYSTEM LIBRARY	92001-13301	1726
92001-16012	%CR2SY	CORE RESIDENT OPERATING SYS.	92001-13301	1726

(Continued)

SOFTWARE MODULE NUMBERS: 92001B LEVEL 1726 (RTEII)

PAPER TAPE	MODULE	DESCRIPTION	CARTRIDGE	DATE CODE
92001-16013	!2GN00	RTE-II 7900 OFF-LINE GEN.	92001-13303	1631
92001-16014	%AUTOP	AUTO RESTART PROGRAM	92001-13302	1631
92001-16018	!2GNFH	RTE-II FIXED HEAD DISC GEN.	92001-13306	1631
92001-16020	%DVA12	2607/10/13/14/17/18 DRIVER	92062-13303	1534
92001-16026	!2GN05	RTE-II 7905 OFF-LINE GEN.	92001-13303	1631
92001-16027	%4DV05	RTE 2644/45 DRIVER	92062-13302	1650
92001-16028	%0DV05	RTE 2640A DRIVER	92062-13302	1650
92001-16029	%%CMD2	RTE-II COMMAND PROGRAM	92001-13301	1710
92001-16030	%WH?T2	RTE-II WHZAT PROGRAM	92001-13302	1726
92001-16031	%RT2G1	RTE-II ON-LINE GENERATOR PT. 1	92001-13304	1704
92001-16031	%RT2G2	RTE-II ON-LINE GENERATOR PT. 1	92001-13304	1704
92001-18014	&AUTOR	AUTO RESTART SOURCE	92001-13302	1631
92001-18033	&AN2F0	RTE-II 7900 GFATHER ANSW FILE	92001-13307	1631
92001-18034	&AN2F5	RTE-II 7905 GFATHER ANSW FILE	92001-13307	1631
92002-12001	%BMPG1	BATCH MONITOR PROGRAM PART 1	92002-13301	1631
92002-12001	%BMPG2	BATCH MONITOR PROGRAM PART 2	92002-13301	1631
92002-12001	%BMPG3	BATCH MONITOR PROGRAM PART 3	92002-13301	1631
92002-12002	%2SP01	RTE-II SPOOL MONITOR PART 1	92002-13303	1631
92002-12002	%2SP02	RTE-II SPOOL MONITOR PART 2	92002-13303	1631
92002-16006	%BMLIB	BATCH LIBRARY	92002-13302	1631
92002-16010	%EDITR	RTE EDITOR	92002-13302	C
92060-12004	%ASMB	RTE ASSEMBLER	92060-13304	1634
92060-12005	%CLIB	RTE COMPILER LIBRARY	92060-13315	1726
92060-16028	%XREF	CROSS REFERENCE	92060-13304	A
92060-16031	%DVR32	RTE 7905A DISC DRIVER	92062-13305	A
92060-16038	%SWTCH	RTE-II SWITCH PROGRAM	92001-13304	1710
92060-16039	%SAVE	SAVE PROGRAM	92060-13309	1704
92060-16040	%RESTR	RESTORE PROGRAM	92060-13309	1704
92060-16041	%VERFY	DISC VERIFY PROGRAM	92060-13309	1704
92060-16042	%COPY	DISC COPY PROGRAM	92060-13309	1704
92060-16043	%DBKLB	DISC BACK UP LIBRARY	92060-13309	1704
92060-16044	!DSKUP	OFF LINE DISC BACK UP	92060-13309	1704
92060-16045	%RDNAM	READ NAME PROGRAM	92001-13302	1631
92060-16052	%KEYS	SOFT KEY UTILITY	92001-13002	1707
92060-16053	%KYDMP	SOFT KEY DUMP UTILITY	92001-13002	1707
92060-16092	%FTN4	RTE FORTRAN IV MAIN	92060-13316	1726
92060-16093	%FFTN4	RTE FORTRAN IV SEG F	92060-13316	1726
92060-16094	%0FTN4	RTE FORTRAN IV SEG 0	92060-13316	1726
92060-16095	%1FTN4	RTE FORTRAN IV SEG 1	92060-13316	1726
92060-16096	%2FTN4	RTE FORTRAN IV SEG 2	92060-13316	1726
92060-16097	%3FTN4	RTE FORTRAN IV SEG 3	92060-13316	1726
92060-16098	%4FTN4	RTE FORTRAN IV SEG 4	92060-13316	1726
92060-18046	&UPDAT	UPDATE TRANSFER FILE	92001-13302	1631
92060-18047	&PKDIS	PACK DISC TRANSFER FILE	92001-13302	1631
92202-16001	%DVR23	RTE 7970 9T. MAG. TAPE DRIVER	92062-13304	A
92900-16002	%2DV47	RTE 92900A DRIVER WITHOUT DMS	92062-13302	1643

SOFTWARE MODULE NUMBERS: 92060B LEVEL 1726 (RTE-III)

The following modules are also available on a 7900 RTE Master Software Disc (#92060-13001), or a 7905 RTE Master Software Disc (#92060-13201), or a 7920 RTE Master Software Disc (#92060-13201).

PAPER TAPE	MODULE	DESCRIPTION	CARTRIDGE	DATE CODE
02607-16004	!S4L07	24K STD LINE PRINTER DRIVER	92001-13305	1538
09601-16021	%DVR15	RTE 7261A DRIVER	92062-13304	A
12732-16001	%DVR33	FLEXIBLE DISC DRIVER	92062-13304	1726
12970-16004	!S4MT1	24K STD MAG.TAPE DRIVER	92001-13305	1550
20747-60001	%DVR30	RTE FIXED HEAD DISC DRIVER	92062-13305	C
20808-60001	%CAL10	CAL. PLOTTER DRIVER	92062-13302	B
20810-60001	%CALIB	CAL. PLOTTER LIBRARY	92062-13302	C
20875-60001	%1FTN	FORTRAN MAIN CONTROL	92060-13308	E
20875-60002	%2FTN	FORTRAN PASS 1	92060-13308	E
20875-60003	%3FTN	FORTRAN PASS 2	92060-13308	E
20875-60004	%4FTN	FORTRAN PASS 3	92060-13308	E
20875-60005	%5FTN	FORTRAN PASS 4	92060-13308	F
24129-60001	%ALGOL	RTE/DOS ALGOL PART 1	92060-13305	1643
24129-60002	%ALGL1	RTE/DOS ALGOL PART 2	92060-13305	C
24153-60001	%FF.N	RTE/DOS FORMATTER	92060-13303	C
24306-60001	%DFCAR	DOSM ST ARITH PK	92060-13303	A
24998-16001	%RLIB1	RTE/DOS LIBRARY PART 1	92060-13302	1726
24998-16001	%RLIB2	RTE/DOS LIBRARY PART 2	92060-13302	1726
24998-16002	%FF4.N	FORTRAN IV FORMATTER	92060-13303	1726
25117-60499	%DVR24	RTE 7970 7T MAG. TAPE DRIVER	92062-13305	D
29013-60001	%DVR31	RTE 7900A DISC DRIVER	92062-13305	1710
29028-60002	%DVR12	RTE 2767A DRIVER	92062-13303	A
29029-60001	%DVR00	RTE TTY/PUNCH/PHOTO READER	92062-13302	1642
29030-60001	%DVR11	RTE 2892A CARD READER DRIVER	92062-13303	1710
29100-60017	!S4LP	24K STD LINE PRINTER	92001-13305	A
29100-60018	!S4SYD	24K STD SYSTEM DUMP	92001-13305	A
29100-60019	!S4PHP	24K STD PHOTO READER	92001-13305	A
29100-60020	!S4PUN	24K STD TAPE PUNCH	92001-13305	A
29100-60022	!S4L67	24K STD 2767 LINE PRINTER	92001-13305	A
29100-60023	!S4MT2	24K STD 7970 MAG. TAPE	92001-13305	A
29100-60049	!S4MT3	24K STD MAG. TAPE	92001-13305	A
29100-60050	!S4TER	24K STD TERMINAL PRINTER	92001-13305	A
59310-16002	%1DV37	RTE HP-IB WITHOUT SRQ	92062-13304	1726
59310-16003	%2DV37	RTE HP-IB WITH SRQ	92062-13304	1726
59310-16004	%HPIB	HP-IB DEVICE SUBROUTINE	92062-13304	1710
59310-16005	%SRQ.P	SRQ.P TRAP UTILITY	92062-13304	1710
72008-60001	%1DV10	COMP. 7210A PLOTTER DRIVER	92062-13302	A
72009-60001	%2DV10	MIN. 7210A PLOTTER DRIVER	92062-13302	A
91200-16001	%DVA13	91200A DRIVER	92062-13303	1648
91200-16002	%TVLIB	91200A VIDEO MONITOR LIBRARY	92062-13303	1648
91200-16004	%TVVER	91200A TV INTERFACE VERIFIER	92062-13303	1648
92001-16003	%MTM	MULT. TERMINAL MONITOR	92060-13301	B
92001-16005	%SYLIB	RTE SYSTEM LIBRARY	92060-13301	1726
92001-16014	%AUTOR	AUTO RESTART PROGRAM	92060-13310	1631
92001-16020	%DVA12	2607/10/13/14/17/18 DRIVER	92062-13303	1534
92001-16027	%4DV05	RTE 2644/45 DRIVER	92062-13302	1650

(Continued)

SOFTWARE MODULE NUMBERS: 92060B LEVEL 1726 (RTE III)

PAPER TAPE	MODULE	DESCRIPTION	CARTRIDGE	DATE CODE
92001-16028	%0DV05	RTE 2640A DRIVER	92062-13302	1650
92001-18014	&AUTOR	AUTO RESTART PROGRAM SOURCE	92060-13310	1631
92002-12001	%BMPG1	BATCH MONITOR PROGRAM PART 1	92002-13301	1631
92002-12001	%BMPG2	BATCH MONITOR PROGRAM PART 2	92002-13301	1631
92002-12001	%BMPG3	BATCH MONITOR PROGRAM PART 3	92002-13301	1631
92002-16006	%BMLIB	BATCH LIBRARY	92002-13302	1631
92002-16010	%EDITR	RTE EDITOR	92002-13302	C
92060-12001	%3SP01	RTE-III SPOOL MONITOR PART 1	92060-13313	1631
92060-12001	%3SP02	RTE-III SPOOL MONITOR PART 2	92060-13313	1631
92060-12003	%CR3SY	MEMORY RESIDENT SYSTEM	92060-13301	1726
92060-12004	%ASMB	RTE ASSEMBLER	92060-13304	1639
92060-12005	%CLIB	RTE COMPILER LIBRARY	92060-13315	1726
92060-16001	%3DP43	POWER FAILURE DRIVER	92060-13301	1633
92060-16004	%LDR3	RTE-III LOADER	92060-13301	1726
92060-16006	%WHZT3	RTE-III WHZAT PROGRAM	92060-13310	1726
92060-16028	%XREF	CROSS REFERENCE	92060-13304	A
92060-16029	!3GN00	7900 RTE-III GENERATOR	92060-13311	1631
92060-16031	%DVR32	RTE 7905A DISC DRIVER	92062-13305	A
92060-16032	!3GN05	7905 RTE-III GENERATOR	92060-13311	1631
92060-16035	%$PVMP	$PVMP	92060-13301	A
92060-16036	%$CMD3	RTE-III COMMAND PROGRAM	92060-13301	1710
92060-16037	%RT3G1	RTE-III ON-LINE GENERATOR PT.1	92060-13312	1704
92060-16037	%RT3G2	RTE-III ON-LINE GENERATOR PT.2	92060-13312	1704
92060-16038	%SWTCH	RTE-III SWITCH PROGRAM	92060-13312	1710
92060-16039	%SAVE	SAVE PROGRAM	92060-13309	1704
92060-16040	%RESTR	RESTORE PROGRAM (RSTOR)	92060-13309	1704
92060-16041	%VERFY	DISC VERIFY PROGRAM	92060-13309	1704
92060-16042	%COPY	DISC COPY PROGRAM	92060-13309	1704
92060-16043	%DBKLB	DISK BACK UP LIBRARY	92060-13309	1704
92060-16044	!DSKUP	OFF LINE DISK BACK UP	92060-13309	1704
92060-16045	%RDNAM	READ NAME PROGRAM	92060-13310	1631
92060-16052	%KEYS	SOFT KEY UTILITY	92062-13310	1707
92060-16053	%KYDMP	SOFT KEY DUMP UTILITY	92060-13310	1707
92060-16092	%FTN4	RTE FORTRAN IV MAIN	92060-13316	1726
92060-16093	%FFTN4	FORTRAN IV SEGMENT F	92060-13316	1726
92060-16094	%0FTN4	FORTRAN IV SEGMENT 0	92060-13316	1726
92060-16095	%1FTN4	FORTRAN IV SEGMENT 1	92060-13316	1726
92060-16096	%2FTN4	FORTRAN IV SEGMENT 2	92060-13316	1726
92060-16097	%3FTN4	FORTRAN IV SEGMENT 3	92060-13316	1726
92060-16098	%4FTN4	FORTRAN IV SEGMENT 4	92060-13316	1726
92060-18046	&UPDAT	UPDATE TRANSFER FILE	92060-13310	1631
92060-18047	&PKDIS	PACK DISK TRANSFER FILE	92060-13310	1631
92060-18050	&AN3F0	RTE-III 7900 GFATHER ANSW FILE	92060-13314	1726
92060-18051	&AN3F5	RTE-III 05/20 GFATHER ANS FILE	92060-13314	1726
92202-16001	%DVR23	RTE 7970 9T. MAG. TAPE DRIVER	92062-13304	A
92900-16002	%2DV47	RTE 92900A DRIVER WITHOUT DMS	92062-13302	1726

SOFTWARE MODULE NUMBERS: 92062B LEVEL 1726 (RTE-III)

PAPER TAPE	MODULE	DESCRIPTION	CARTRIDGE	DATE CODE
09601-16021	%DVR15	RTE 7261A DRIVER	92062-13304	A
12732-16001	%DVR33	FLEXIBLE DISC DRIVER	92062-13304	1726
20747-60001	%DVR30	RTE FIXED HEAD DISC DRIVER	92062-13305	C
20808-60001	%CAL10	CAL. PLOTTER DRIVER	92062-13302	B
20810-60001	%CALIB	CAL. PLOTTER LIBRARY	92062-13302	C
25117-60499	%DVR24	RTE 7970 7T MAG. TAPE DRIVER	92062-13305	D
29013-60001	%DVR31	RTE 7900A DISC DRIVER	92062-13305	1710
29028-60002	%DVR12	RTE 2767A DRIVER	92062-13303	A
29029-60001	%DVR00	RTE TTY/PUNCH/PHOTO READER	92062-13302	1642
29030-60001	%DVR11	RTE 2892A CARD READER DRIVER	92062-13303	1710
59310-16002	%1DV37	RTE HP-IB WITHOUT SRQ	92062-13304	1726
59310-16003	%2DV37	RTE HP-IB WITH SRQ	92062-13304	1726
59310-16004	%HPIB	HP-IB DEVICE SUBROUTINE	92062-13304	1710
59310-16005	%SRQ.P	SRQ.P TRAP UTILITY	92062-13304	1710
72008-60001	%1DV10	COMP. 7210A PLOTTER DRIVER	92062-13302	A
72009-60001	%2DV10	MIN. COMP. 7910A PLOTER DRIVE	92062-13302	A
91200-16001	%DVA13	91200A DRIVER	92062-13303	1648
91200-16002	%TVLIB	91200A VIDEO MONITOR LIBRARY	92062-13303	1648
91200-16004	%TVVER	91200A TV INTERFACE VERIFIER	92062-13303	1648
92001-16020	%DVA12	2607/10/13/14/17/18 DRIVER	92062-13303	1534
92001-16027	%4DV05	RTE 2644/45 DRIVER	92062-13302	1650
92001-16028	%0DV05	RTE 2640A DRIVER	92062-13302	1650
92060-16031	%DVR32	RTE 7905A DISC DRIVER	92062-13305	1704
92202-16001	%DVR23	RTE 7970 9T. MAG. TAPE DRIVER	92062-13304	A
92900-16002	%2DV47	RTE 92900A DRIVER WITHOUT DMS	92062-13302	1643
92900-16003	%3DV47	RTE 92900A DRIVER WITH DMS	92062-13302	1643

SOFTWARE MODULE NUMBERS: 92064A OPTIONS 20 & 40 LEVEL 1726 (RTE-M)

The following modules are unique in that they are available on Flexible disc as well as Paper Tape and Mini-Cartridge.

STRUCTURE

The RTE-M operating system is divided into three groups. Refer to the RTE-M Programmer's Reference Manual (part no. 92064-90002) for a description of the operating systems.

Within this list the modules that correspond with each operating system are described as MI, MII, or MIII.

CARTRIDGE TAPES

There are three cartridge tapes that contain the three operating systems. The part numbers of these cartridge tapes and the corresponding operating systems follow:

92064-13301	RTE-MI
92064-13302	RTE-MII
92064-13303	RTE-MIII

Modules that correspond with two or all three operating systems and are contained on more than one cartridge tape contain (MI), (MII), or (MIII) in their description.

Modules that do not directly relate to the operating systems are contained on the other cartridge tapes.

FLEXIBLE DISCS

There are two flexible discs referred to as GEN DISC and APP DISC. The GEN DISC (92064-13401) contains all the software that can be loaded at generation. The APP DISC (92064-13402) contains all the application software that can be loaded on-line. As with the cartridge tapes, some of the modules can be found on both flexible discs.

The Generation disc contains the following:

- Off-line generator
- All operating system software
- I/O drivers
- Certain HP user programs

The Applications disc contains the following:

- HP applications programs — Assembler
  FORTRAN compiler
  Editor
  Cross reference program
- Certain relocatable system software
- Certain user programs

Modules that appear on both flexible discs contain (GEN DISC) or (APP DISC) in their description.

SOFTWARE MODULE NUMBERS: 92064A OPTIONS 20 & 40 LEVEL 1726 (RTE-M)

PAPER TAPE	MODULE	DESCRIPTION	CARTRIDGE	FLEXIBLE DISC	DATE CODE
09601-16021	%DVR15	RTE 7261A CARD READER DRIVER	92062-13304	92064-13401	A
12732-16001	%DVR33	FLEXIBLE DISC DRIVER	92062-13304	92064-13401	1650
20808-60001	%CAL10	RTE PLOTTER DRIVER	92062-13302	92064-13401	B
20810-60001	%CALIB	CAL. PLOTTER LIBRARY	92062-13302	92064-13401	C
24153-60001	%FF.N	RTE/DOS FORTRAN FORMATTER	92060-13303	92064-13401	C
24153-60001	%FF.N	RTE/DOS FORTRAN FORMATTER	92060-13303	92064-13402	C
24306-60001	%DECAR	DOSM STRING ARITH PK	92060-13303		A
24998-16001	%RLIB1	RTE/DOS LIBRARY	92060-13302	92064-13402	1624
24998-16001	%RLIB1	RTE/DOS LIBRARY	92060-13302	92064-13401	1624
24998-16001	%RLIB2	RTE/DOS LIBRARY	92060-13302	92064-13401	1624
24998-16001	%RLIB2	RTE/DOS LIBRARY	92060-13302	92064-13402	1624
24998-16002	%FF4.N	FORTRAN IV FORMATTER	92060-13303	92064-13402	1624
24998-16002	%FF4.N	FORTRAN IV FORMATTER	92060-13303	92064-13401	1624
29028-60002	%DVR12	RTE 2767A DRIVER	92062-13303	92064-13401	A
29029-60001	%DVR00	RTE TTY/PUNCH/PHOTO READER	92062-13302	92064-13401	1642
29030-60001	%DVR11	RTE 2892A CARD READER DRIVER	92062-13303	92064-13401	1710
59310-16002	%1DV37	HP-IB WITHOUT SYSTEM REQUEST	92062-13304	92064-13401	1710
59310-16003	%2DV37	HP-IB WITH SYSTEM REQUEST	92062-13304	92064-13401	1710
59310-16004	%HPIB	HP-IB RTE UTILITY	92062-13304	92064-13401	1710
59310-16005	%SRQ.P	SRQ.P TRAP UTILITY	92062-13304	92064-13401	1710
72008-60001	%1DV10	COMP. 7210A PLOTTER DRIVER	92062-13302	92064-13401	A
72009-60001	%2DV10	MIN. COMP. 7210A PLOTTER DRIVE	92062-13302	92064-13401	A
91200-16001	%DVA13	91200 TV INTERFACE DRIVER	92062-13303	92064-13401	1648
91200-16002	%TVLIB	VIDEO MONITOR LIBRARY	92062-13303	92064-13401	1648
91200-16004	%TVVER	TV INFT VERIF	92062-13303	92064-13401	1648
92001-16020	%DVA12	2607/10/13/14/17/18 DRIVER	92062-13303	92064-13401	1534
92001-16027	%4DV05	RTE 2644/45 DRIVER	92062-13302	92064-13401	1650
92001-16028	%0DV05	RTE 2640A DRIVER	92062-13302	92064-13401	1650
92060-16052	%KEYS	SOFT KEY UTILITY	92064-13304	92064-13402	1707
92060-16053	%KYDMP	SOFT KEY DUMP UTILITY	92064-13304	92064-13402	1707
92060-16092	%FTN4	FORTRAN IV MAIN		92064-13402	1726
92060-16093	%FFTN4	RTE FORTRAN IV SEG ID SUB		92064-13402	1726
92060-16094	%0FTN4	FORTRAN IV SEGMENT 0		92064-13402	1726
92060-16095	%1FTN4	FORTRAN IV SEGMENT 1		92064-13402	1726
92060-16096	%2FTN4	FORTRAN IV SEGMENT 2		92064-13402	1726
92060-16097	%3FTN4	FORTRAN IV SEGMENT 3		92064-13402	1726
92060-16098	%4FTN4	FORTRAN IV SEGMENT 4		92064-13402	1726
92064-12005	%FMPC	CARTRIDGE FMP/FMPCR (LIB)	92064-13306	92064-13401	1709
92064-12006	%FMPF	FLEX DISC FMGR LIB (GEN DISC)		92064-13401	1726
92064-12006	%FMPF	FLEX DISC FMGR LIB (APP DISC)		92064-13402	1726
92064-12007	%CLIBM	RTE COMPILER LIBRARY		92064-13402	1726
92064-16001	%MSY1	MI OPERATING SYSTEM	92064-13301	92064-13401	1726
92064-16002	%MSY2	MII OPERATING SYSTEM	92064-13302	92064-13401	1726
92064-16003	%MSY3	MIII OPERATING SYSTEM	92064-13303	92064-13401	1726
92064-16005	%MBU	MI BUFFERING	92064-13301	92064-13401	1650
92064-16006	%MMP	MI SCHEDULING OPTION	92064-13301	92064-13401	1650

(Continued)

SOFTWARE MODULE NUMBERS: 92064A OPTIONS 20 & 40 LEVEL 1726 (RTE-M)

PAPER TAPE	MODULE	DESCRIPTION	CARTRIDGE	FLEXIBLE DISC	DATE CODE
92064-16008	%MTI	TIMER OPTION (MI)	92064-13301	92064-13401	1650
92064-16008	%MTI	TIMER OPTION (MIII)	92064-13303	92064-13401	1650
92064-16008	%MTI	TIMER OPTION (MII)	92064-13302	92064-13401	1650
92064-16009	%MTS	TIME SCHEDULING OPTION (MIII)	92064-13303	92064-13401	1650
92064-16009	%MTS	TIME SCHEDULING OPTION (MI)	92064-13301	92064-13401	1650
92064-16009	%MTS	TIME SCHEDULING OPTION (MII)	92064-13302	92064-13401	1650
92064-16010	%MOP	OPERATOR COMMAND OPTION (MIII)	92064-13303	92064-13401	1650
92064-16010	%MOP	OPERATOR COMMAND OPTION (MI)	92064-13301	92064-13401	1650
92064-16010	%MOP	OPERATOR COMMAND OPTION (MII)	92064-13302	92064-13401	1650
92064-16011	%MCL	CLASS I/O OPTION (MII)	92064-13302	92064-13401	1726
92064-16012	%MAP	MI/II ABSOLUTE PROGRAM LOADER	92064-13305	92064-13401	1726
92064-16013	%MDMLB	DUMMY LIBRARY (MII)	92064-13302	92064-13401	1650
92064-16013	%MDMLB	DUMMY LIBRARY (MIII)	92064-13303	92064-13401	1650
92064-16013	%MDMLB	DUMMY LIBRARY (MI)	92064-13301	92064-13401	1650
92064-16015	%MCL3	CLASS I/O OPTION (MIII)	92064-13303	92064-13401	1726
92064-16016	%MAP3	MIII ABSOLUTE PROGRAM LOADER	92064-13305	92064-13401	1726
92064-16017	%FMGC0	CARTRIDGE FILE MANAGER	92064-13305	92064-13401	1709
92064-16018	%DRC	CARTRIDGE DIR MAN PROGRAM	92064-13304	92064-13401	1650
92064-16019	%TBLCR	CARTRIDGE DIRECTORY TABLES	92064-13304	92064-13401	1650
92064-16021	%DRC1	MI CARTRIDGE DIRECTORY SUBR	92064-13306	92064-13401	1650
92064-16022	%RTMGN	SYSTEM GENERATOR	92064-13305	92064-13401	1726
92064-16023	%RTMLD	RELOCATING LOADER (APP DISC)	92064-13305	92064-13402	1726
92064-16023	%RTMLD	RELOCATING LOADER (GEN DISC)	92064-13305	92064-13401	1726
92064-16024	%RTMSC	LOADER SUB CONTROL (GEN DISC)	92064-13305	92064-13401	1726
92064-16024	%RTMSC	LOADER SUB CONTROL (APP DISC)	92064-13305	92064-13402	1726
92064-16025	%MEDIT	EDITOR		92064-13402	1703
92064-16026	%MASM6	CROSS REFERENCE SEGMENT		92064-13402	1650
92064-16027	%MPF	MI/II POWER FAIL	92064-13304	92064-13401	1650
92064-16029	%MPF3	MIII POWER FAIL	92064-13304	92064-13401	1650
92064-16030	%MAUTO	AUTOR REL	92064-13304	92064-13401	1650
92064-16031	%MRN	RESOURCE NUMBER MANAGER (MII)	92064-13302	92064-13401	1650
92064-16031	%MRN	RESOURCE NUMBER MNGR (MIII)	92064-13303	92064-13401	1650
92064-16032	%ONMTM	MULTI TERMINAL MONITOR (GEN D)	92064-13305	92064-13401	1650
92064-16032	%ONMTM	MULTI TERMINAL MONITOR (APP D)	92064-13305	92064-13402	1650
92064-16033	!MCGEN	ABSOLUTE CARTRIDGE GENERATOR	92064-13307		1726
92064-16034	%SGPRP	SEGMENT PROGRAM PREP		92064-13402	1650
92064-16035	%MPRMP	PROMPT (MTM)	92064-13305	92064-13401	1650
92064-16036	%MRSPN	RESPONSE (MTM)	92064-13305	92064-13401	1650
92064-16040	%MASM0	ASSEMBLER MAIN CONTROL		92064-13402	1650
92064-16041	%MASM1	ASSEMBLER SEGMENT 1		92064-13402	1650
92064-16042	%MASM2	ASSEMBLER SEGMENT 2		92064-13402	1650
92064-16043	%MASM3	ASSEMBLER SEGMENT 3		92064-13402	1650
92064-16044	%MASM4	ASSEMBLER SEGMENT 4		92064-13402	1650
92064-16045	%MFTN0	FORTRAN MAIN CONTROL		92064-13402	1650
92064-16046	%MFTN1	FORTRAN SEGMENT 1		92064-13402	1650
92064-16047	%MFNT2	FORTRAN SEGMENT 2		92064-13402	1650
92064-16050	%MASM5	ASSEMBLER SEGMENT 0		92064-13402	1650
92064-16051	%MXRF0	CROSS REFERENCE MAIN		92064-13402	1650
92064-16054	%DIRD	CARTRIDGE DIRECTORY READ	92064-13304	92064-13401	1650
92064-16055	%FMGF0	FLEX DISC FILE MNGR (GEN DISC)		92064-13401	1709
92064-16055	%FMGF0	FLEX DISC FILE MNGR (APP DISC)		92064-13402	1709
92064-16056	%DPF	F DISC DIRECT PROG (GEN DISC)		92064-13401	1650
92064-16056	%DPF	F DISC DIRECT PROG (APP DISC)		92064-13402	1650
92064-16057	%TBLFP	FLEXIBLE DISC DIRECT TABLES		92064-13401	1709
92064-16060	%DRF1	F DISC DIRECTORY SUB (GEN D)		92064-13401	1650
92064-16060	%DRF1	F DISC DIRECTORY SUB (APP D)		92064-13402	1650
92064-16075	!MFGEN	ABSOLUTE FLEXIBLE DISC SYSTEM		92064-13401	1726
92064-16080	%STRTM	RTE-M SYSTEM START-UP	92064-13304	92064-13401	1709
92064-16081	%MSYLB	RTE-M SYSTEM LIBRARY (GEN DISC	92064-13306	92064-13401	1709
92064-16081	%MSYLB	RTE-M SYSTEM LIBRARY (APP DISC	92064-13306	92064-13402	1709
92064-18059	&TBLCR	CARTRIDGE DIRECTORY TBLS SOURC	92064-13306	92064-13402	1650
92064-18126	&MHELP	EDITOR HELP FILE SOURCE		92064-13402	1650
92064-18141	&MAUTO	AUTOR SOURCE	92064-13306	92064-13402	1650
92064-18171	&TBLFP	FLEXIBLE DISC DIRECTORY SOURCE		92064-13402	1709
92202-16001	%DVR23	RTE 7970 9T. MAG. TAPE DRIVER	92062-13304	92064-13401	A
92900-16002	%2DV47	RTE 92900A DRIVER WITHOUT DMS	92062-13302	92064-13401	1643
92900-16003	%3DV47	RTE 92900A DRIVER WITH DMS	92062-13302	92064-13401	1643

TRAINING SCHEDULE

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 1977 through August 1977.

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 imrnediately in order that we may offer your seat to another student.

TRAINING CENTER ADDRESSES

Cupertino

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

Sunnyvale

974 East Arques
Sunnyvale, California

Rockville

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

Boise

P.O. Box 15
15 N. Phillippi Street
Boise, Idaho 83707
(208) 376-6000
TWX: 910-970-5784

Boblingen

Kundenschulung
Herrenbergerstrasse 110
D-7030 Boblingen, 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

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

TITLE

TRAINING COURSE RATES AND CENTER LOCATION

Course Number	Length	Price	Cupertino	Sunnyvale	Rockville	Boise	Boblingen	Winnersh	Grenoble	Milan	*** Madrid	Stockholm	*** Amsterdam/ Brus.
01ETC	RTE II/III Driver Writing Course				Nov 30								
	3 days	300											
22940A	2100 Maint.			Sep 12 Oct 10 Nov 7 Dec 5									
	10 days	$1000											
22941A	21MX Maint.			Sep 26 Oct 3 Oct 24 Oct 31 Nov 28					Dec 5				
	5 days	500											
22942A	7900 Maint.			Sep 26 Oct 3 Oct 24 Oct 31 Nov 28					Nov 28				
	5 days	500											
22943A	7970B Maint.					Sep19 Nov14							
	5 days	600											
22944A	7970E Maint.					Sep12 Nov7							
	5 days	600											
22945A	7905 Maint.			Sep 12 Sep 19 Oct 10 Oct 17 Nov 7 Dec 5 Dec 12					Nov 14				
	5 days	500											
22950A	2100 Ser. Assm.		Sep 19 Oct 10 Nov 7 Dec 5		Sep 26 Oct 3 Nov 7 Dec 5		Oct 10 Nov 21 Jan 30	Oct 10	Sep 26 Dec 12	Nov14 Jan23		Oct 10 Nov 28	
	5 days	500											
22952A	DOS III B		**										
	5 days	500											
22960A	21MX Mic. Prog.		**										
	5 days	500											
22965B	RTE-II/III		Sep 12 Sep 19 Sep 26 Oct 3 Oct 10 Oct 17 Oct 17 Oct 24 Oct 24 Oct 31 Nov 7 Nov 14 Dec 5 Dec 12		Sep 12 Sep 19 Oct 10 Oct 17 Oct 24 Oct 31 Nov 7 Nov 14 Nov 28 Dec 5 Dec 12 Dec 19		Sep 26 Oct 3 Oct 24 Nov 7 Nov 28 Dec 5 Jan 9 Jan 16	Sep 26 Oct 3 Nov 14 Nov 21	Oct 3 Oct17 Nov 21 Dec 5 Jan 9 Jan 23	Sep19 Oct3 Nov28 Dec19		Oct 17 Oct 24 Dec 5 Dec 12	
	10 days	1000											
	(Course includes RTE-II/III operating system, batch spool monitor and file manager.)												
22968A	Measurement & control		**						Oct 17				
	2 days	200											

TITLE — TRAINING COURSE RATES AND CENTER LOCATION

Course Number	Length	Price	Cupertino	Sunnyvale	Rockville	Boise	Boblingen	Winnersh	Grenoble	Milan	*** Madrid	Stockholm	*** Amsterdam/ Brus.
22969A	Distr. Sys.		Oct 24 Nov 28		Oct 3		Oct 17		Nov 28				
	5 days	500											
22977A	Image/DBMS 1000		Sep 12 Oct 31 Dec 12		Sep 26				Oct 10 Jan 16	Oct 17 Jan9			
	5 days	500											
22978	TCS		**										
	2 days	200											
22979A	Real/Time Multiterminal Basic		**						Oct 19				
	3 days	300											
22980B	HPIB Minicomputer Environment		Oct 3 Nov 28						Nov 7				
	4 days	400											
22983A	21MX E-Micro-programming		Sep 26 Oct 17 Nov 14 Dec 12						Oct 24				
	5 days	500											
22984A	7920 Maint.			Sep 19 Nov 14									
	5 days	500											
22985A	RTE-M		Oct 3 Nov 7 Dec 5		Nov 14				Dec 12				
	5 days	500											

*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. Course 22977A, IMAGE/DBMS 1000 replaces 22953A (2100 IMAGE); the new class adds additional material and extends the training from 3 to 5 days.

**On Sufficient Demand.

***We have not yet received dates for Madrid or Amsterdam/Brus.

HEWLETT-PACKARD
COMPUTER SYSTEMS COMMUNICATOR ORDER FORM

Please Print:

Name _____ Title _____

Company _____

Street _____

City _____ State _____ Zip Code _____

Country _____

☐ HP Employee Account Number _____ Location Code _____

☐ **DIRECT SUBSCRIPTION**

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

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

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

☐ **SERVICE CONTRACT CUSTOMERS**

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

Number of additional copies _____

FOR HP USE ONLY

CONTRACT KEY

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

Approved _____

HEWLETT-PACKARD
COMMUNICATOR SUBSCRIPTION AND ORDER INFORMATION

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

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

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

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

SAMPLE

☒ **DIRECT SUBSCRIPTION**

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

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

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

SAMPLE

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

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

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

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

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

HEWLETT-PACKARD
LOCUS CONTRIBUTED SOFTWARE CATALOG
DIRECT MAIL ORDER FORM

Please Print:

Name _____ Title _____

Company _____

Street _____

City _____ State _____ Zip Code _____

Country _____

☐ HP Employee Account Number _____ Location Code _____

Part Number	Description	Qty.	List Price Each	Extended Total
22000-90099	Locus Contributed Software Catalog		$15.00	

*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*	
Handling Charge	1.50
TOTAL	

Domestic Customers: Mail the order form with your check or money order (payable to Hewlett-Packard Co.) to:

HEWLETT-PACKARD COMPANY
LOCUS CATALOG
P.O. Box 61809
Sunnyvale, CA 94088

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

NOTE: No direct mail order can be shipped outside the United States. All prices domestic U.S.A. only. Prices are subject to change without notice.

Please photocopy this order form if you do not want to cut the page off. You will automatically receive a new order form with your order.

HEWLETT hp PACKARD

CONTRIBUTED SOFTWARE
Direct Mail Order Form

NOTE: No direct mail order can be shipped outside the United States.

Please Print:

Name _____ **Title** _____

Company _____

Street _____

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

Country _____

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

Sub-total		
Your State & Local Sales Taxes*		
Handling Charge	1	50
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.

Domestic Customers: Cash required on all orders less than $50.00. Mail the order form with your check or money order (payable to Hewlett-Packard Co.) or your U.S. Company Purchase Order to:

HEWLETT-PACKARD COMPANY
Contributed Software
P.O. Box 61809
Sunnyvale, CA 94088

International Customers: Order through your local Hewlett-Packard Sales office. No direct mail order can be shipped outside the United States.

All prices domestic U.S.A. only. Prices are subject to change without notice.

ORDERING INFORMATION FOR LOCUS CONTRIBUTED SOFTWARE

Programs are available individually in source language on either paper tape, magnetic tape, or cassettes as indicated in the abstracts.

To order a particular program, it is necessary to specify the program identification number, together with an option number which indicates the type of product required. The program identification number with the option number composes the ordering number.

For example:

22113A-K01

The different options are:

K01 — Source paper tape and documentation
K21 — Magnetic tapes and documentation

NOTE

Specify 800 BPI or 1600 BPI Magnetic tape.

B01 — Binary tape and documentation
D00 — Documentation
L00 — Listing

Not all options are available for all programs.

Ten-digit numbers do not require additional option numbers such as K01, K21, etc. The 10-digit number automatically indicates the option or media ordered.

For example:

22681-18901 — The digits 189 indicate source paper tape plus documentation.
22681-10901 — The digits 109 indicate source magnetic tape plus documentation (800 BPI magnetic tape)
22681-11901 — The digits 119 indicate source magnetic tape plus documentation (1600 BPI magnetic tape)
22681-13301 — The digits 133 indicate source cassettes plus documentation

Only those options listed in each abstract are available.

Refer to the Price List for prices and correct order numbers.

Hewlett-Packard offers no warranty, expressed or implied and assumes no responsibility in connection with the program material listed.

GUIDELINES FOR CONTRIBUTED SOFTWARE

New contributed programs are welcome for consideration as entries to the Data Systems (LOCUS), this is HP's opportunity to expand communication among users. A new contributor of an accepted program receives a traditional plaque engraved with his/her name, and the choice of any other contributed program of equal dollar value. This guide contains the necessary information for contributing all Assembly language, Fortran, and Algol programs, and also microprograms, which will run on an HP 2100, 21MX Series computer.

The Contributed Library compiles and/or assembles but does not test or maintain contributed entries. Program maintenance is the responsibility of the person submitting the program because he/she is the most knowledgeable concerning the entry. The Contributed Library does assume responsibility, however, for collecting and forwarding any error reports and/or incomplete documentation reports. (A software report form is included at the end of this guide and one is distributed with each program ordered from the Library.)

Minimum Program Package

"Proper documentation" for possible acceptance into the Contributed Library includes: machine readable source tapes (documentation should be contained in source code when possible), Program Documentation Form (included in this guide) which is to be typed and in reproducible form, Contributed Library Disclaimer (included in this guide), and a test case.

Documentation

The Documentation Form becomes an important reference to understanding and using a contributed program; therefore, it is important that the directions be clear and specific. When filling out this form, the contributor may encounter subdivisions requesting information inappropriate to his/her contributed package. In such instances the contributor should respond with "NONE" or "N/A" (not applicable) to insure that important documentation is not missed. Although the documentation form has been shortened, do not omit information needed to operate the program. The contributor can easily expand on any section of the Documentation Form by adding plain white 8-1/2" × 11" paper, and copying the section number followed by the word "continued".

NOTE

All documentation, including sample runs, core allocation lists, and flow charts will be commercially printed. Therefore, all items must be submitted dark enough and clear enough to conform to printer standards. This applies especially to all computer out-output.

Test Case

It is important that the user be able to verify proper functioning of the program he/she has received (i.e., no punch errors, bugs, etc.). The contributor is asked to provide a set of test inputs for the program and the corresponding output. This may take the form of a "debug" program, a listing of data values, a data source tape with accompanying output list, or in the case of contributed subroutines and microprograms, any small program that demonstrates its proper function. If this test case policy is not applicable to the program, the contributor may use the Example Input/Output Test Case Section of the Documentation Form to discuss how a user should verify that he/she has received and loaded a program properly.

Mailing Procedures

All program packages submitted to the Hewlett-Packard Data Systems (LOCUS) should be wrapped securely and sent to:

Hewlett-Packard Data Systems (LOCUS)
11000 Wolfe Road
Cupertino, Calif. 95014

Maintenance

The Hewlett-Packard Software Center approach to program maintenance has found general agreement among users as the only way to keep the quality of contributed programs at a high level of performance. Program maintenance implies correction of program errors, and clarification of program documentation, most often accomplished by revisions to a program. To be effective, the Contributed Library has to insure the integrity of its programs; therefore, no response to program errors may result in the removal of the program from the library.

Revisions

There are many possible reasons for program revisions, including: clarification of documentation, correction of program "bugs", added versatility to a program, and any minor changes in coding designed to improve program efficiency. Revision must always be accompanied by a new Contributed Library Disclaimer checked as a revision. Only those items which are changed need be resubmitted.

Replacement

A program replacement differs from a revision in that it is a completely recorded program meant to perform the same or similar function of some previously submitted package. Replacements are handled as new entries and are assigned new program identification numbers. Suggestions for upgrading existing programs are appreciated and will be carefully reviewed by Contributed Library personnel.

CONTRIBUTED PROGRAM DOCUMENTATION FORM FOR HEWLETT-PACKARD DATA SYSTEMS (LOCUS)

PROGRAM NUMBER/TITLE

PROGRAM LANGUAGE(s)

[ ] FORTRAN IV
[ ] FORTRAN II
[ ] ALGOL
[ ] RELOCATABLE ASSEMBLY LANGUAGE
[ ] ABSOLUTE ASSEMBLY LANGUAGE
[ ] MICROPROCESSOR ASSEMBLY LANGUAGE

PROGRAM TYPE

[ ] PROGRAM
[ ] DRIVER
[ ] OTHER _____
[ ] SUBROUTINE
[ ] FUNCTION

SUPPORTED SOFTWARE REQUIREMENTS - OPERATING SYSTEM

[ ] DOS-M
[ ] BCS
[ ] MTS
[ ] RTE
[ ] SIO SYSTEM
[ ] HP BASIC VERSION _____
[ ] SELF-CONTAINED
[ ] OTHER _____

PROGRAM DESCRIPTION

CONTRIBUTED PROGRAM DOCUMENTATION FORM (2)

CONTRIBUTED SOFTWARE REQUIREMENTS

(Name and order number of other contributed packages required)

HARDWARE REQUIREMENTS

TAPE IDENTIFICATION

Tape #	Contents	Control Statement (if any)

LOAD AND RUN INSTRUCTIONS

CONTRIBUTED PROGRAM DOCUMENTATION FORM (3)

DIAGNOSTIC MESSAGES/ADDITIONAL EXITS

SUBROUTINES OR MICROPROGRAMS

LITERATURE REFERENCE

SPECIAL CONSIDERATIONS

EXAMPLE INPUT/OUTPUT (TEST CASE)

HEWLETT-PACKARD DATA SYSTEMS (LOCUS) PROGRAM DISCLAIMER

[ ] INITIAL SUBMISSION

☐☐☐ Classification Code

Up to 4 Cross Reference Words

1. _____ 2. _____

3. _____ 4. _____

[ ] REVISION (PROGRAM NO. _____)

[ ] Errors Corrected

[ ] Documentation Clarified

[ ] Other reason _____

_____

TITLE

CONTRIBUTOR'S NAME

ORGANIZATION

ADDRESS

TELEPHONE

Name and organization to appear in catalog? [ ] yes [ ] no

May a user contact you directly? [ ] yes [ ] no

MATERIAL SUBMITTED

[ ] Documentation [ ] Source Program [ ] Test Case

[ ] Other _____

DISCLAIMER:

To the best of my knowledge, this contributed program is free of any proprietary information belonging to any person or organization. I am making this program information available to the HEWLETT-PACKARD CONTRIBUTED LIBRARY. I hereby agree that HP may reproduce, publish, and use it, and authorize others to do so without obligations or liability of any kind.

_____ _____

(Signature) (Date)

HP DATA SYSTEM (LOCUS)
CONTRIBUTED SOFTWARE REPORT

Use the space provided to comment on any problems or modifications on HP 2100 contributed software and enclose any output that may be useful. A copy will be forwarded to the contributor, and a reply will be returned to the person who submits this report. Send completed report to:

Hewlett-Packard Data Systems (LOCUS)
11000 Wolfe Road
Cupertino, California 95014

Submitted By _____ Date _____

Organization Name _____ Program Name _____

Address _____ Order No. _____

City, State, Zip _____

Phone _____

Has software been modified by user? NO _____ YES _____ (If YES, explain below)

Enclosed References:

TTY Log _____ Listing _____ Corrected Tape _____ Corrected Listing _____

Although every effort is made to ensure 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.

Printed in U.S.A. 8/77

Part No. 5951-6111