

RSTS PROFESSIONAL

Volume 4, Number 6

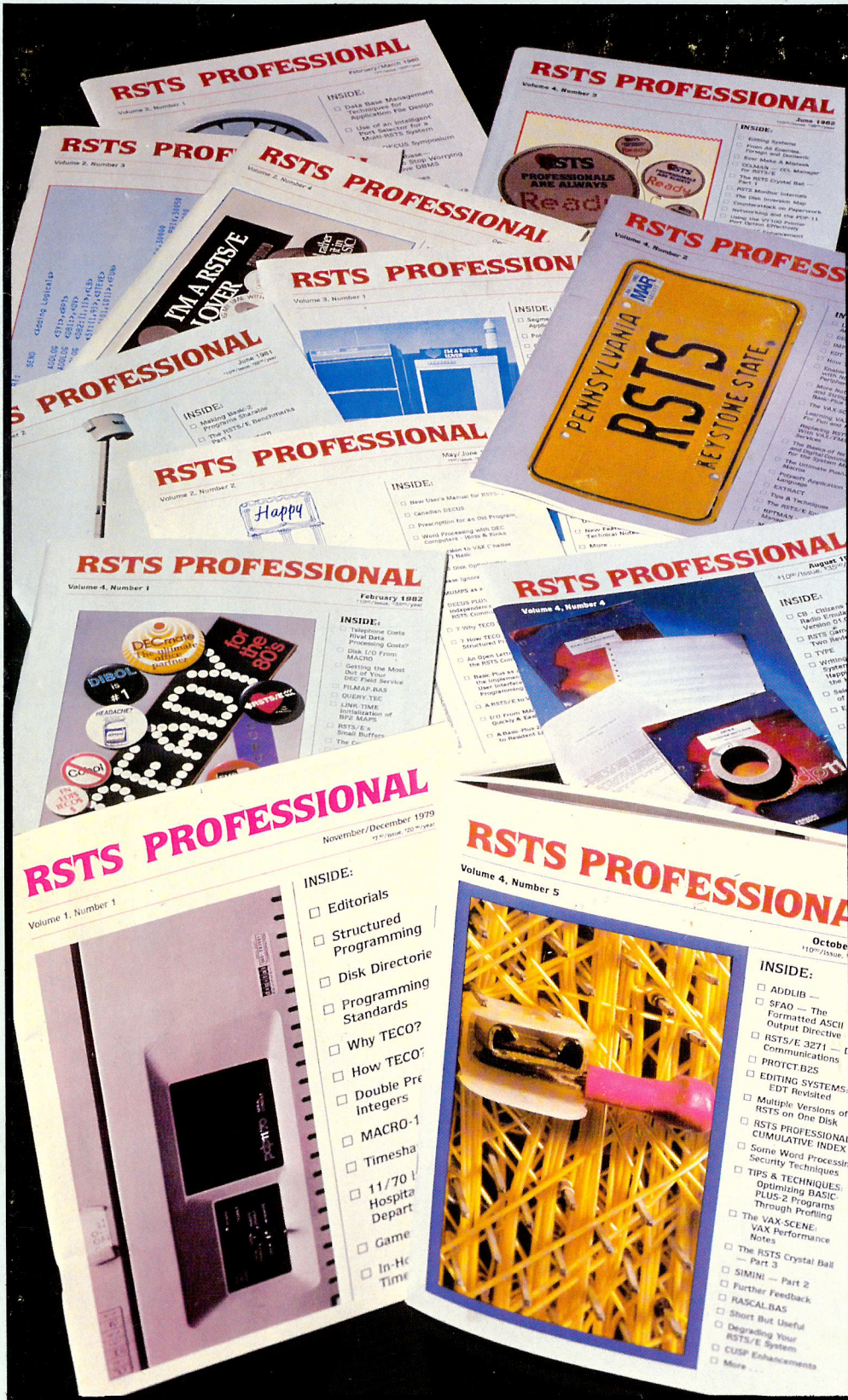
Third Anniversary Issue

December 1982

\$10⁰⁰/issue, \$35⁰⁰/year

INSIDE:

- ☐ How Do You Read RSTS/E Monitor Tables
- ☐ USRDSK — Disk Usage Summary Report for RSTS/E
- ☐ EDITING SYSTEMS: Some EDT Wishes Granted
- ☐ FORMS
- ☐ More on Using the VT100 Printer Port
- ☐ ATPK.TEC
- ☐ WRTOUT.BAS
- ☐ PROFIL
- ☐ The VAX-SCENE: Reading RMS Files From VAX MACRO
- ☐ TIPS & TECHNIQUES: RSTS Job Context
- ☐ HOW TECO
- ☐ SIMINI — Part 3
- ☐ A User Written Keyboard Monitor
- ☐ CACHE FLOW
- ☐ A Security Checklist
- ☐ DECUS Australia
- ☐ More . . .





USER-11: POWERFULLY PRODUCTIVE.

People productivity. It's more important than ever. And a good database system can mean *real* productivity.

USER-11 is a high-performance database system.

It is a fact: Software designed with USER-11 is built more quickly, operates more reliably, and performs better than other software techniques.

USER-11 is unique. It's easy to install. Easy to learn. And easy to apply. Adaptive tools and a standard approach ensure that maintenance is easier than ever.

A key to USER-11's success is its powerful, dictionary-based modules. Software developers

Naturally USER-11 is supported with excellent documentation and a variety of training options for beginner to expert. Our commitment is to your complete satisfaction.

Whether you are a software provider or a software user, we guarantee you will be delighted.

Ask us about USER-11 and our family of business software products, or better yet, ask a *productive* USER!



North County
Computer Services, Inc.
2235 Meyers Ave.,
Escondido, California 92025

What's the most efficient way to distribute financial models from DEC* to desks?

MAPS/Host™ & MAPS/Pro™



* DEC, VAX and PDP are registered trademarks of Digital Equipment Corporation.
MAPS, MAPS/Host and MAPS/Pro are registered trademarks of Ross Systems Incorporated.

**OK,
ROSS**

... I'd like to know more about how MAPS can offer me true
distributed financial planning.

Please send more information on ☐ MAPS/Host ☐ MAPS/Pro

☐ I/We have access to a VAX or PDP-11

☐ Have a Ross representative call on me

Name _____

Title _____

Company _____

Address _____

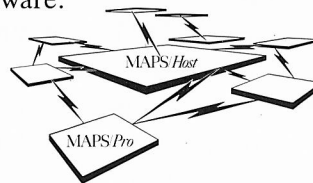
City _____ State _____ Zip _____

Phone () _____

MAPS™ financial modeling software lets you offer your decision makers a single, comprehensive solution to their complex financial processing needs.

MAPS/Pro software is designed to run on DEC's powerful new Professional 350 desktop computer—to give you the full benefit of the 350's P/OS multi-tasking operating system, 5MB Winchester disk, bit-mapped graphics, special function keys and application menus. Fully-compatible MAPS/Host software runs on PDP-11 and VAX computers.

Working independently at desktop 350s, MAPS lets you and your users take advantage of state-of-the-art microprocessing hardware and software.



With 350s linked together, MAPS' common planning language and centrally-administered data base allow users to exchange data and financial models directly. Plus users can upload applications to a PDP-11 or VAX when additional computing power is needed—making *truly* distributed financial planning possible.

Designed for total financial decision support, MAPS features unlimited logic, professional report formats, sophisticated consolidation capabilities and advanced data calculations. Plus instant access to a common data base and model library that maximizes user efficiency. With MAPS there's no limit to the size or complexity of the system you can develop. Yet MAPS' online Help, Business English, and visual data editing make MAPS easy to use. And, fully-documented, it's easy to learn. What's more, Ross "Hot-line" support is as close as the phone if a problem does occur.

For more information on MAPS software, just return our coupon. Or call Ross Systems toll free at (800) 547-1000 (in California, call (415) 856-1100).



1860 Embarcadero Road
Palo Alto, CA 94303
Regional offices in San Francisco,
New York, Dallas, Los Angeles.

SOFTWARE

Provo, UT — Passive security deters tamperers. Familiar methods involve passwords, multi-level access privileges, terminal locks, account dedicated terminals, and other access restrictions.

Need for Passive Security. Passive security provides a necessary deterrent to unauthorized access. Operating systems like RSTS/E and VMS come with a degree of passive security already built in. Additional hardware and software can increase the level of passive security.

Not Enough. Passive security by itself is not enough. A determined individual can break into even the most protected computer systems. Once a user has gained access to the system, passive security weakens. It can't see what the user is doing at the terminal. After gaining access to the system, a covert user works in almost total privacy.

The Active Security Approach. A bank locks its money in a vault and protects its tellers with bullet proof glass. Despite this, it still provides for the occasional undeterred thief. Cameras visually record activities in the bank. Here a covert user does not work in privacy. Illegal activity is recorded as it occurs. This is active security.

A Software Camera. Just as a bank needs a video monitor, so does a computer. But the type of camera needed is different. The computer needs a software camera which records all activity occurring at a user's terminal.

SYSTEM SECURITY

PASSIVE

—VS—

ACTIVE

CONTRL — Interactive Software Camera. With CONTRL you can look at what is happening on any terminal on your system. You see at your terminal exactly what a remote or local user is typing and exactly what the computer is answering back to the user. With CONTRL watching, a covert user no longer works in private.

Secured Record. While watching with CONTRL, you can request that CONTRL record all of the user's activities in a log file. This log represents a legal history of what the user did at the terminal.

Invisible Eye. The user is unaware that CONTRL is watching. CONTRL does not interfere in any way with programs the user runs.

MONITR — Automatic Software Camera. For fully automated active security you may use MONITR. Like CONTRL it watches and records all activity on a user's terminal. However, MONITR runs without your intervention.

Programmed Monitoring. You tell MONITR which terminals and users to monitor on a periodic basis. MONITR then runs detached and does as instructed. Later you can review MONITR's recorded log for

suspicious terminal activity. Because of MONITR's flexibility, you can program it to give as tight a level of active security as desired.

Simple Installation. CONTRL and MONITR run under any RSTS/VMS system. They require no changes to the operating system or to the terminal handler. Just drop the software cameras on your system and they work immediately. In fact they don't even require film. Complete documentation is included with the distribution kit.

Clyde Digital Systems, Inc. is the principle supplier of system utilities and application programs for the PDP-11 and VAX.

The logo for Clyde Digital Systems, Inc. features the word "Clyde" in a stylized, bold, orange-red font. The letter "C" is particularly large and has a unique shape, with the "l" and "y" following in a similar bold, sans-serif style.

How To Get More Information. Call Lisa at (801) 224-5306, or write **Clyde Digital Systems, Inc.**, 370 North Canyon Road/Bldg. 3-E Provo, UT 84604.

Contents

HOW DO YOU READ RSTS/E MONITOR TABLES?	8
Michael H. Koplitz	
PEEK, PEEK, PEEK.	
Here are the definitive peek sequences you always wanted.	
USRDSK — Disk Usage Summary Report for RSTS/E	16
Philip Hunt	
Remember DSKLST, well here's one better.	
EDITING SYSTEMS — Some EDT Wishes Granted	22
David Spencer	
David keeps making EDT better and better. Happy holidays from our Editor Editor.	
FORMS	26
Jeff Corbett	
While all IBM operators are skilled at forms alignment, we can use the help.	
MORE ON USING THE VT100 PRINTER PORT	30
Terry Ridgers	
How to get more from your computer ports: having a VT100 and a printer share one.	
The idea just keeps getting better.	
ATPK.TEC	32
Steve Roy	
Want to have command substitution with ATPK? TECO can do it!	
WRTOUT.BAS	34
Ray Tomlinson	
I did this years ago, but someone always has a better way.	
PROFIL	38
Joe Blaho	
We have talked about "profiling" in earlier issues. Here is an implementation in BP2.	
THE VAX-SCENE	
READING RMS FILES FROM VAX MACRO	48
Bob "MACRO MAN" Meyer	
Reading RMS files directly can be fun. It can also be done in Basic Plus, just ask	
Enterprise Technology. Macro Man uses MACRO of course.	
TIPS & TECHNIQUES — RSTS Job Context	50
Wef Fleischman	
Internals can also be fun, 'specially if it's from the experts at Software Techniques.	
Wef explains what a job actually looks like inside RSTS.	
HOW TECO	56
Andrew Harrison	
Is CATALOG finally obsolete? How you can do it in TECO.	
SIMINI — A Simulation Mini Computer - PART 3	58
John Cato	
Here are the sources and some final words on the series.	
A USER WRITTEN KEYBOARD MONITOR	59
Ken Harris	
Would you like users NEVER to have ↑c work? Your own keyboard monitor and	
default run-time system could do it.	
CACHE FLOW	63
Kevin Paul Herbert	
Want to know how the RSTS Cache works, just ask Kevin.	
Someone did so Kevin found out all about it and now he tells us. I'm glad Kevin is	
directed towards the forces of good and RSTS.	
A SECURITY CHECKLIST	74
Dave Mallery	
DECUS AUSTRALIA	75
Carl B. Marbach	
VERSION 7.2 OF RSTS/E NOW AVAILABLE	80
LATE BREAKING NEWS	81

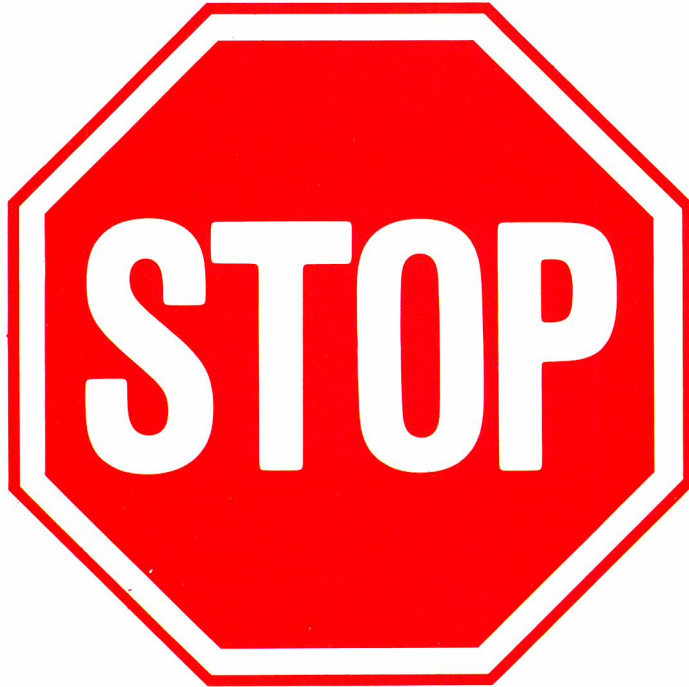
Coming . . .

- Run-Time Systems Explained (again)
- Number Systems for the PDP-11
- START.BAS
- DIBOL and MACRO, Together!
- QUEDEV.B2S
- V7.2 Reports
- DEXPO and DECUS Revisited
- New Security System Features
- How to Backup a RSTS System
- Floppy Disks Explained
- A Query Language in BASIC PLUS and Record I/O
- More . . .

From the Editors	4
Letters to the RSTS Pro	6
Dear RSTS Man	36
News Releases	82
Classifieds	86
List of Advertisers	87

The RSTS Professional Magazine, December 1, 1982, Vol. 4, No. 6. Published bi-monthly. Single copy price \$10⁰⁰, \$35⁰⁰ per year; \$50 Canada and 1st class. All other countries, air mail, \$60 US. Published by M Systems, Inc., 161 E. Hunting Park Avenue, Philadelphia, Pa. 19124. Application to mail at Second Class postage rates is pending at Philadelphia, Pa. POSTMASTER: send all correspondence and change of address to: RSTS Professional, Box 361, Ft. Washington, Pa. 19034-0361, telephone (215) 542-7008. Copyright © 1982 by M Systems, Inc. No part of this publication may be reproduced in any form without written permission from the publisher.

BEFORE you add memory (or anything else) to increase system performance



You should add DOPTER!

DOPTER is an easy to use RSTS/E disk copying program which

INCREASES SYSTEM PERFORMANCE UP TO 50%.

DOPTER performs all of the standard functions necessary to structure a RSTS/E disk volume and automatically does the following:

- Places all files and free space in their optimum positions.
- Produces better optimized MFD/UFD's than REORDR.
- Deletes unused file attributes from source, task, and object library files saving UFD and cache accesses.
- Places and pre-extends the MFD.
- Places the most used files at the front of the UFD's.
- Places the UFD's with the most activity toward the front of the MFD.

For More Information

If you would like more information on how you can increase the performance of your RSTS/E system with DOPTER and a free copy of "RSTS/E DISK OPTIMIZATION IN A MULTI-USER ENVIRONMENT", phone or write SPH today.

RSTS/E is a registered trademark of Digital Equipment Corporation.



**System
Performance
House, Inc.**

5522 Loch More Court • Dublin, Ohio 43017 • 614-265-7788

CIRCLE 108 ON READER CARD

LETTERS to the RSTS Pro...

Send letters to: Letters to the RSTS Pro, P.O. Box 361, Ft. Washington, PA 19034-0361.

RMS temporary files opened under BASIC +2 take on a default system name in Version 7.0 of RSTS; usually of the form: TEMPnn.TMP. Basic +2 under RMS will ignore the account and file name that you specify in your OPEN statement. Problems can arise when you run a program that opens a temporary file on a specific DISK PACK drive from a different account; say a privileged program run from a non-privileged account. If that non-privileged account was not created on that specific DISK PACK drive, then you will receive a system error: ?Disk PACK IS PRIVATE.

roy Gorka, Systems & Computer Services, DuPont Canada Inc., Mississauga, Ontario, has come across a possible RSTS Version 7.0 BASIC +2 RMS System Bug. It appears that if two files are opened on the same channel (even if the first one is closed before the second OPEN statement is executed), then BASIC +2 under RMS will expect the second file to be located on the same device as the first file, if there is no explicit device specified for the second file. It does not default to the system disk.

Terry Ridgers, Analyst/Programmer
DuPont Canada Inc.
St. Clair River Works, Corunna, Ont.

Since I started reading (and enjoying) the *RSTS Professional* about two years ago, I've seen various articles about program preprocessors (BLINK, Dec, 1981), program standards, maintenance philosophies, etc.

As expected, everybody has their own methods, and does their own things, in their own little world.

If in fact there are some 10,000 RSTS sites in the world, it would seem to me that consolidation of these 9,999 various methods of program maintenance and development would be a worthwhile effort.

A BASIC-Plus-2 future, combined with an ever-increasing maintenance load, is forcing us to review the BASIC-Plus pre-processor that we currently use.

I'd like to issue a friendly challenge to your readers to send me their ideas on preprocessors. I will summarize the input I receive and report the results in a future issue of *RSTS Professional*. For discussion, some features we have or are considering are:

- non-standard SCALE factors
- conditional assembling
- compile-time variables
- labels vs. line numbers
- generation of .ODL files
- nesting of command files
- source module libraries
- batch compilation
- ability to process ANY language text or command file
- compilation instructions internal to a

program

Please forward your notes, suggestions, examples to:

Garry D. Robbins, Technical Services
Analyst
Labatt Brewing Company, 150 Simcoe St.
London, Ontario N6A 4M3
Phone (519) 673-5186

P.S. Keep up the excellent work!

Don't forget that BASIC V2 will soon be upon us with some of these features already built in (labels for instance). Maybe the pre-processor needs to work for BASIC, BASIC V2, BASIC-Plus-2 and VAX-BASIC.

The *RSTS Pro* proved invaluable last week when after a disk failure we were left with a dirty pack with a user data error in the UFD for “1,3”. With the aid of Vol. 2, No. 4 we reset the “mounted” bit, set the pointer to the 1st name entry in the UFD to zero, and were then able to proceed with recovering the disk. Wonderful.

There are some important things going on at DEC that the commercial SIG ought to be pursuing. Did you know, for example, that the CTS500 release of the DIBS Dibol accounting application software will use DMS, not RMS? I think that the whole point of the SIG should be to prevent idiocies such as this on DEC's part. Oh well...

Looking forward to hearing from you.

Rob Lamb, System Manager
LA Computer Services Ltd, London

I wanted to compliment you on a fine magazine. The days crawl from issue to issue and I am looking forward to the time "RSTS Pro" becomes a monthly magazine.

Have there been any articles on how to interpret the ERRDIS listing?

It appears to be a useful diagnostic tool, but other than the date and heading I can't seem to interpret anything intelligible. (And DEC doesn't seem to be interested in its results either.)

I have enclosed a little silliness that I put together for a laugh. I hope you enjoy it. Keep up the good work.

Dennis DeMaria
Data Processing Manager
Ark-Les Corp., Watertown, Mass.

Thank you, Dennis. Readers can find your "sillyness," **Chéz RSTS** on page 24.

Integrated Services, Inc., writes thousands of checks and other documents from our PDP computers each month which require the writing of dollars and other numbers in English, with both check and new program volume increasing monthly. Consequently we needed a routine which would convert a number into a string of words without using much program space since we are currently using BASIC PLUS.

I accomplished this with the attached function routine which will create such a

string with a value up to .1E 13 -.01, which, so far, exceeds our needs. It would be a simple matter to increase the maximum amount by increments of 1000 times the current limit. The routine uses a minimum of string space overhead and minimum of code by looping back into itself after finding and parsing the largest "set" within the number (i.e., millions) passed to it.

My intentions are to remove the function (FN) part of the routine when it is converted to a CALLED routine in BASIC PLUS II, then placed in a general usage library available to all programs on the system. Having also written FORTRAN and COBOL, I feel this code should be easily converted to one of these languages as a COPY for COBOL and subroutine for FORTRAN.

Raymond D. Tomlinson
Systems Programmer

Integrated Services, Inc., Aurora, CO.
See Raymond's article "WRTOUT.BAS" on
page 34 of this issue.

I am writing on behalf of the DECUS Australia Chapter Board and members of the RSTS SIG to express our thanks and appreciation for your services which helped make our 1982 Symposium the success it was.

Both during and since the Symposium I have been gathering comments from attendees to include in my report to the Chairman of the Symposium Planning Committee. All comments associated with the pre-symposium seminar you conducted have been most favourable and it looks as though it is going to be a very hard act to follow in future symposia.

I would like also to take this opportunity to express my appreciation for the very active role you played in the main part of our Symposium in the days that followed your seminar.

We hope you enjoyed your stay in Australia as much as we enjoyed your being here and that you have pleasant memories of your visit.

Yours faithfully,
Chris Brett
DECUS Australia

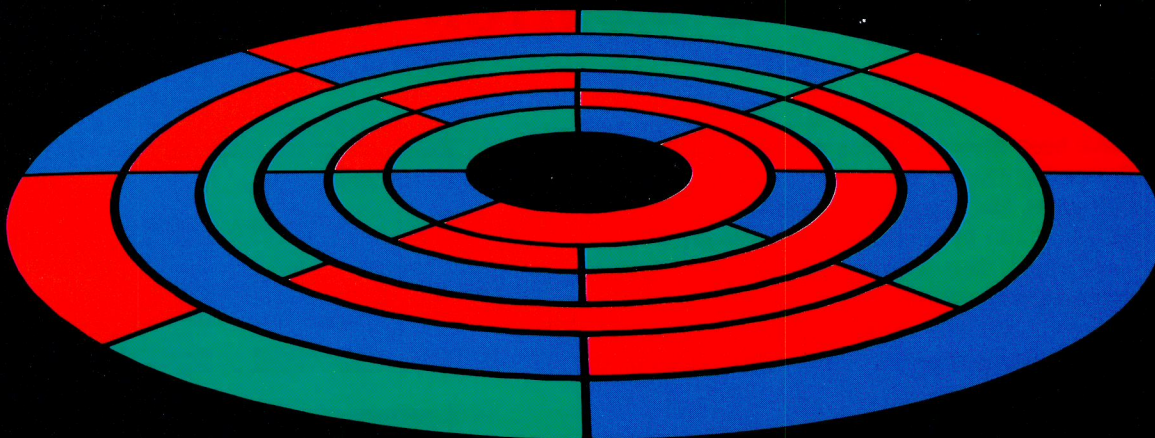
I used to love TECO. I have followed the many articles in the RSTS PROFESSIONAL on TECO with great interest. It is nice to know that someone still appreciates the only true 'DEC Standard Editor'. But, there are always two sides to any story, and I hope you will forgive me for bringing up some anti-TECO points.

I am enclosing, unedited, a copy of a letter I wrote to the editor of a DECUS SIG newsletter a few months ago. I don't think it was published. Perhaps you might want to publish it, if for no other reason than to show that the RSTS PROFESSIONAL dares to print the opinions of those whose beliefs run contrary to those of the majority.

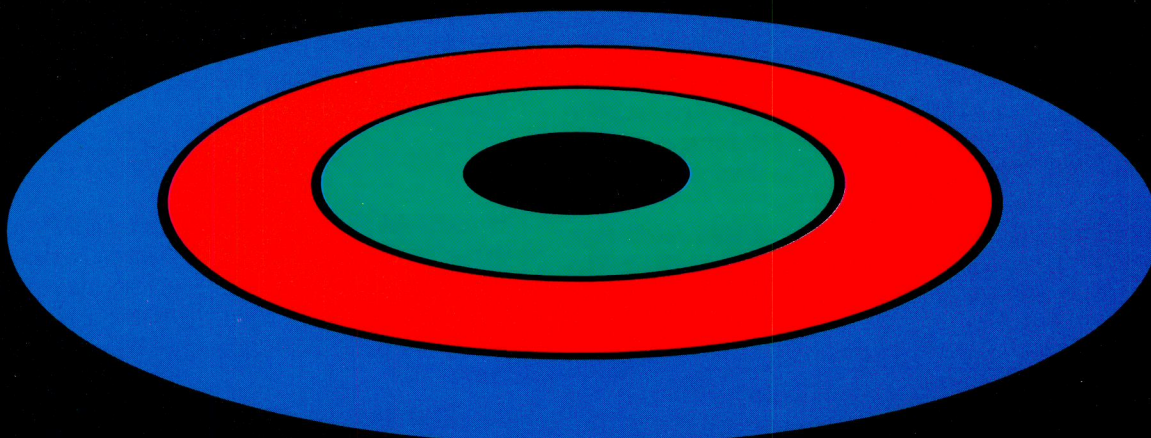
Michael E. Mazzoni, President
Process Control Systems, Inc.
Brookfield, WI

... continued on page 40

WHAT YOU DON'T KNOW ABOUT YOUR DISKS IS COSTING YOU MONEY



If your disk looks like this, you're wasting system performance.



If your disk looks like this, you're using DISKIT.

When the job you're running requires reading the "red" file, it naturally happens faster on a well-ordered disk. Disks become "fragmented" as you use your computer. The system slows down. And that costs you money.

Now, you can restructure your disks and get back that lost performance (up to 50%) without spending a dime on new hardware. DISKIT is the original software system that makes this possible.

But don't confuse DISKIT with other system utilities, DISKIT is a complete "software tool kit" that optimizes your RSTS/E system.

DISKIT is:

- DSU — The utility which restructures the information on your disk, making data fast and easy to access.
- DIR — The incredible directory tool that finds files at the rate of 400 per second.
- RDR — Reorders disk directories 30 times faster than ever before possible.
- OPEN — Displays complete job statistics and file activity so you can see what your system is doing.
- DUS — The set of CALLable subroutines which pre-extend file directories, reducing fragmentation.

In today's tight economy, it's more important than ever to get the most out of your hardware investment. Call or write today and start getting your money's worth from your computer.

Software
Techniques
Incorporated

5242 Katella Avenue
Los Alamitos, CA 90720
United States
Phone: (714) 995-0533

287 London Road
Newbury, Berkshire RG13 2QJ
United Kingdom
Phone: 44 (0) 635-30840

CIRCLE 65 ON READER CARD

HOW DO YOU READ RSTS/E MONITOR TABLES?

By Michael H. Koplitz

There have been several articles written that deal with the RSTS/E monitor tables, but none of these articles explains how to write the algorithms necessary to read the tables. BASIC-PLUS can be used to read the monitor tables instead of MACRO-11. This article will give the algorithms necessary to read any table in the job data structure for any job. It is assumed that the reader has an understanding about the job data structures (all of the 1981 issues of the RSTS Pro contain articles about the tables) and of BASIC-PLUS.

One SYS call will be used which gives the pointers to the job table. The job table contains pointers to the individual JDBs, the first job data structure (figure A gives the job data structures).

```
JOBTL% = SWAP%(CVT$(MID(SYS(CHR$(6%)+CHR$(-3%)),11%,2%)))
```

Every memory location can now be accessed by the PEEK function. The variable JOBTBL% points to the first location in the job table. The first entry of the job table is for job zero, the null job, this entry is zero. The final entry of the table is -1.

When utilizing SYS calls the reader should recall that the job number is always returned as JOB.NUMBER*2. The reason for this is that the entries in the job table increase by words.

```

-----
!job 0 JDB pointer !JOBTBL%
-----
!job 1 JDB pointer !JOBTBL%+(JOB.NUMBER%*2%)
-----
!job 2 JDB pointer !JOBTBL%+(JOB.NUMBER%*2%)
-----
!
!
!
-----
!      -1      !
-----
job table

```

Therefore the JDB% starting address for any job is:

```
JDB% = PEEK(JOBTBL%+(JOB.NUMBER%*2%))
```

If JDB% is zero then the job does not exist on the system.

The entries in the job table can be gathered by the following algorithm:

```

JDI0B% = PEEK(JDB%)
JDPLG% = PEEK(JDB%+2%)
JDI0ST% = PEEK(JDB%+4%) AND 255%
JDPOST% = SWAP%(PEEK(JDB%+4%)) AND 255%
JDI0RK% = PEEK(JDB%+6%)
JDI0B2% = PEEK(JDB%+8%)
JDPLG2% = PEEK(JDB%+10%) AND 255%
JDI0ZN% = SWAP%(PEEK(JDB%+10%)) AND 255%
JDI0TS% = PEEK(JDB%+12%)
JDI0SQ% = PEEK(JDB%+14%)
JDI0ZE% = PEEK(JDB%+22%) AND 255%
JDI0RI% = PEEK(JDB%+28%) AND 255%
JDI0RST% = SWAP%(PEEK(JDB%+28%)) AND 255%
JDI0WAP% = SWAP%(PEEK(JDB%+30%)) AND 255%

```

The priority of the job, JDPRI% can be negative. In twos complement the negative number will display as a positive number because it is only a byte and the seventh bit becomes the sign bit. Therefore the following statement is performed if the value is negative:

```

JDPRI% = JDPRI% - 256%
IF JDPRI% > 128%

```

This statement moves the sign bit from bit 7 to bit 15, resulting in a negative number.

The words of the secondary block job, JDB2 can now be read as follows:

```

JDB2% = JDI0B2%
J2TICK% = PEEK(JDB2%)
J2CPU% = PEEK(JDB2%+2%)
J2CPU% = PEEK(JDB2%+4%)
J2KCT% = PEEK(JDB2%+6%)
J2DEV% = PEEK(JDB2%+8%)
PROGRAM.NAME$ = RAD$(PEEK(JDB2%+12%))
+ RAD$(PEEK(JDB2%+24%))
PROJECT.NUMBER% = SWAP%(PEEK(JDB2%+25%)) AND 255%
PROGRAMMER.NUMBER% = PEEK(JDB2%+24%) AND 255%

```

A pattern can be seen on how to gather the information from the data structures. Only offsets are needed to read any information. If the data is in the high order byte the word must be "SWAP%" first, then "AND 255%" on the swapped word will give the bits. Data in the low order byte can be obtained by the word "AND 255%".

When obtaining the current account in a program the function

```

A% = PEEK(PEEK(PEEK(520%)+8%)+24%)
PROJ% = SWAP%(A%) AND 255%
PROG% = A% AND 255%

```

is used. An analysis of this function shows where this data actually comes from. PEEK(520%), memory location 520 decimal, 1010 octal, contains the starting address of the JDB block for the current running job. Therefore JDB% is equal to PEEK(520%), the starting address of the JDB structure.

The next location would be PEEK(JDB% + 8%), substituting JDB% for PEEK(520%). Note that this location is the pointer to the job's JDB2 structure. Therefore JDB2% is equal to PEEK(JDB% + 8%) and when substituted back into the original function the result is:

```
A% = PEEK(JDB2%+24%)
```

Offset 24 in the JDB2% is the PPN. A minor change to this function would allow the gathering of any account number for any job.

```

A% = PEEK(PEEK(PEEK(JOBTBL%+JOB.NUMBER%*2%)+8%)+24%)
PROJ% = SWAP%(A%) AND 255%
PROG% = A% AND 255%

```

The program running for a job would be:

```

PROGRAM.NAME$ = RAD$(PEEK(PEEK(PEEK(JOBTBL%
+ (JOB.NUMBER%*2%))+8%)+12%))
+ RAD$(PEEK(PEEK(PEEK(JOBTBL%
+ (JOB.NUMBER%*2%))+8%)+14%))

```

The RTS being used is:

```

RTS.NAME$ = RAD$(PEEK(PEEK(PEEK(JOBTBL%
+ (JOB.NUMBER%*2%))+12%)+2%))
+ RAD$(PEEK(PEEK(PEEK(JOBTBL%
+ (JOB.NUMBER%*2%))+12%)+4%))

```

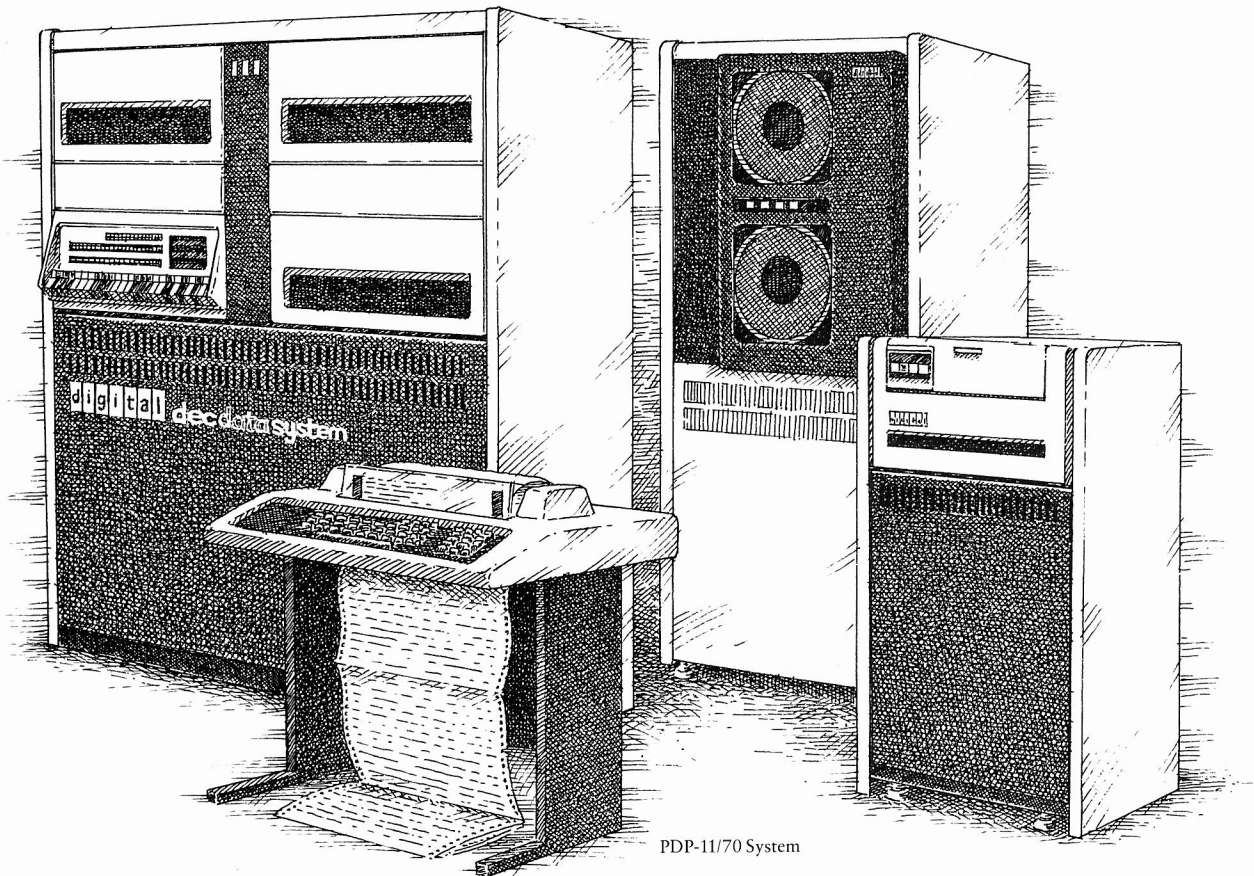
If the job is a receiver then the ID is:

```

ID$ = ""
ID$ = CHR$(SWAP%(PEEK(PEEK(PEEK(JOBTBL%
+ (JOB.NUMBER%*2%))+8%)+18%)+X%)) AND 255%))
+ CHR$(PEEK(PEEK(PEEK(PEEK(JOBTBL%
+ (JOB.NUMBER%*2%))+8%)+18%)+X%)) AND 255%))
FOR X% = 2% TO 6% STEP 2%
IF PEEK(PEEK(JOBTBL%+(JOB.NUMBER%*2%)+8%)+18%)
> 0%

```


State of the Art. Again.



Most computers that have been used are just that. Used. But Digital thinks you deserve better than that. So, before we resell any system, we completely refurbish it at the factory.

We actually modernize it to our latest engineering standards. Even the warranty we give it is state of the art. The same 90-day warranty that goes on our new equipment.

That's the difference between our systems and anybody else's. Everybody's were fresh from the factory once. But ours are fresh from the factory twice.

Call your local Digital sales office or 1-800-258-1728.
In New Hampshire, (603) 884-7240.
Digital Equipment Corporation,
Traditional Products, 16 Hampshire
Drive, Hudson, NH 03051.

digital


```

001 *****
\
\
\ THIS PROGRAM WAS WRITTEN BY M H KOPLITZ, 23-FEB-82
\
\
\ THIS PROGRAM WILL DISPLAY THE JOB DATA STRUCTURES FOR ANY JOB
\
\ *****
010 EXTEND
\
040 PRINT "Job interval";
\ INPUT LINE AS
\ AS = CVT$(AS,4%)
\ GOTO 050 IF AS = ""
\ Y% = INSTR(1%,AS," ")
\ J1% = VAL(AS) IF Y% = 0%
\ J1% = VAL(LEFT(AS,Y%-1%)) IF Y% > 0%
\ J2% = VAL(RIGHT(AS,Y%+1%)) IF Y% > 0%
\ J2% = J1% IF Y% = 0%
\
050 JOBTBL% = SWAP%(CVT$(MID(SYS(CHR$(6%)+CHR$(-3%)),1%,2%))) !GET START ADDRESS OF
! JOB TABLE
\
090 JOB.NUMBER% = JOB.NUMBER% + 1%
\ JOB.NUMBER% = 1%
\ IF PEEK(JOBTBL%
+ (JOB.NUMBER% * 2%)) = -1%
GOTO 090 IF PEEK(JOBTBL% + (JOB.NUMBER%*2%))
= 0%
\ GOTO 100 IF J1% = 0%
\ GOTO 090 IF JOB.NUMBER% < J1%
\ GOTO 090 IF JOB.NUMBER% > J2%
\
!GET NEXT JOB NUMBER
!LOOK AT, WHEN ZERO
!NO JOB, -1 MEANS END
!OF TABLE SO RESET
\
100 PRINT FNC$(0,0);FNC$(0,5);FNC$(1,25);
\ "Job";JOB.NUMBER%;
\ FNC$(0,7);FNC$(0,10);"Data";
\ FNC$(0,9);FNC$(0,8);" Structures";
\ PRINT FNC$(2,15);FNC$(0,7);FNC$(0,10);"JDB";
\ FNC$(0,9);FNC$(0,8);
\ JDBPTR% = PEEK(JOBTBL% + (JOB.NUMBER%*2%)) !PRINT HEADERS, GET
!POINTER TO JDB
\
105 PRINT FNC$(3,1);FNC$(0,7);FNC$(0,25);
\ FNC$(0,10);"1";
\ STRING$(34%,113%);"k"
\ PRINT "x"
\ FOR X% = 4% TO 13%
\ PRINT "m";STRING$(34%,113%);"j";FNC$(0,8)
\ FOR X% = 4% TO 13%
\ PRINT FNC$(X%,36%);FNC$(0,7);FNC$(0,10);
\ FNC$(0,25);"x";FNC$(0,9);
\ FNC$(0,8)
\ NEXT X% IDRAW THE PRETTY BOX
\
110 PRINT FNC$(4,8);"Primary stats: ";
\ JFPOST% = PEEK(JDBPTR% + 2%)
\ PRINT ".RSX";
\ IF JFPOST% AND 2%**2%
\ PRINT ".^C";
\ IF JFPOST% AND 2%**3%
\ PRINT ".2^C";
\ IF JFPOST% AND 2%**4%
\ PRINT ".PR";
\ IF JFPOST% AND 2%**10%
\ PRINT ".TP";
\ IF JFPOST% AND 2%**11%
\ PRINT ".NL";
\ IF JFPOST% AND 2%**12%
\ PRINT ".LK";
\ IF JFPOST% AND 2%**14%
\ PRINT PRIMARY STATS
\
120 PRINT FNC$(5,11);"Error code: ";
\ JDIOST% = PEEK(JDBPTR% + 4%) AND 255%
\ PRINT JDIOST% !PRINT ERROR CODE TO
! RETURNED AFTER
! MONITOR CALL
\
130 PRINT FNC$(6,9);"Posting mask: ";
\ JDPOST% = SWAP%(PEEK(JDBPTR% + 4%)) AND 255%
\ PRINT JDPOST% !PRINT FIRQB POST MA
\
140 PRINT FNC$(7,9);"Second stats: ";
\ JDFLG2% = PEEK(JDBPTR% + 10%) AND 255%
\ PRINT ".MP";
\ IF JDFLG2% AND 2%**1%
\ PRINT ".NP";
\ IF JDFLG2% AND 2%**2%
\ PRINT ".LO";
\ IF JDFLG2% AND 2%**6%
\ PRINT ".KJ";
\ IF JDFLG2% AND 2%**7%
\ PRINT SECONDARY STATS
\
150 PRINT FNC$(8,9);"Swap in size: ";
\ JDSIZN% = SWAP%(PEEK(JDBPTR% + 14%)) AND 255%
\ PRINT JDSIZN% !PRINT THE SWAP SIZE
\
160 PRINT FNC$(9,4);"Residency quantum: ";
\ JDRESQ% = PEEK(JDBPTR% + 14%)
\ PRINT JDRESQ% !PRINT THE RESIDENCY
! QUANTUM
\
170 PRINT FNC$(10,10);"Memory size: ";
\ JDSIZE% = PEEK(JDBPTR% + 22%) AND 255%
\ PRINT JDSIZE% !PRINT THE MEMORY SI
\
180 PRINT FNC$(11,13);FNC$(0,7);FNC$(0,10);
\ "Priority: ";
\ JDPRI% = PEEK(JDBPTR% + 28%) AND 255%
\ JDPRI% = JDPRI% - 256% IF JDPRI% > 128%
\ PRINT JDPRI%;FNC$(0,9);FNC$(0,8);
\ !PRINT THE PRIORITY
\
190 PRINT FNC$(12,13);FNC$(0,7);FNC$(0,10);
\ "Runburst: ";
\ JDBRST% = SWAP%(PEEK(JDBPTR% + 28%)) AND 255%
\ PRINT JDBRST%;FNC$(0,9);FNC$(0,8);
\ !PRINT THE RUNBURST

```

FIGURE A. Some RSTS/E Job Structures



Some interesting things happen when you sit down at our CIT-161 Terminal.

Like any office worker, video terminal operators are bombarded by a tremendous spectrum of visual and audio stimulæ. Add to that their own thought distractions and you begin to see why a single-color display is at a severe disadvantage competing for user attention.

Our CIT-161 Color Alphanumeric Terminal, on the other hand, makes use of some very interesting color phenomena. Which makes a much more efficient terminal... and makes users a lot more productive.

Studies have shown, for instance, that color can improve user response time by 50% — which translates into faster throughput and decreased time to job completion. Data is made more interesting and easy to follow with the use of color, resulting in less error in interpretation.

Color also has broad attention-getting power, affording the added ability to prioritize data by different hues and



create more significant and long-lasting impressions. Not to mention the subliminal conditioned responses we all have to color. (Red, for example, can be used to flag a problem instantly.)

And color dramatically accelerates the learning process. Meaning that training time can be substantially reduced so operators reach full productivity far more quickly.

How will our CIT-161 fit into your system? Perfectly! It's an ANSI-compatible direct replacement for the DEC® VT-100 Series with a full range of advanced video features: 132/80-character column display, single-width or double-width/double-height characters, the unique Window Erase feature, split screen, and more.

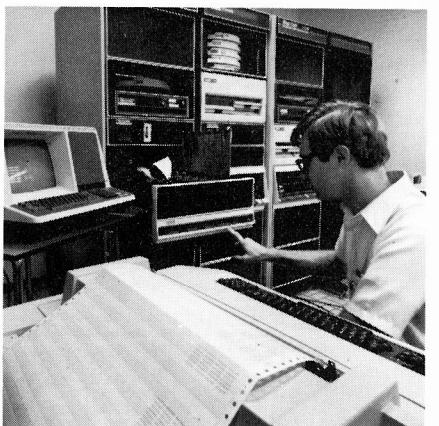
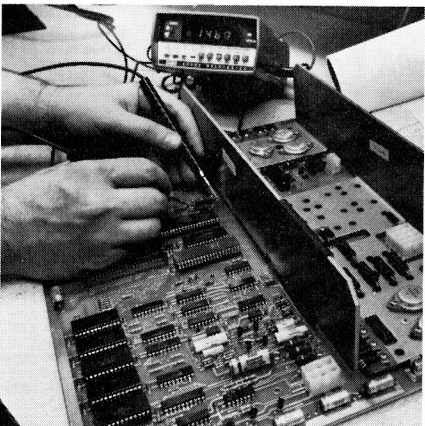
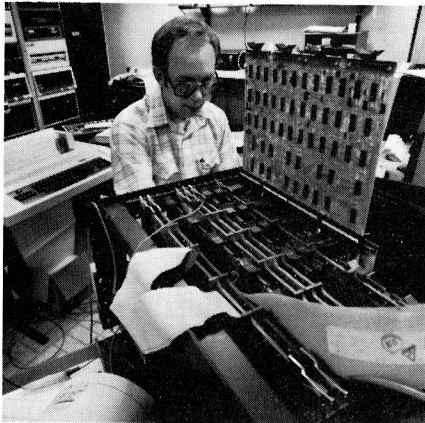
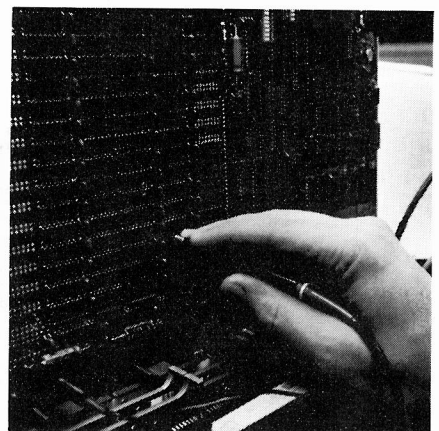
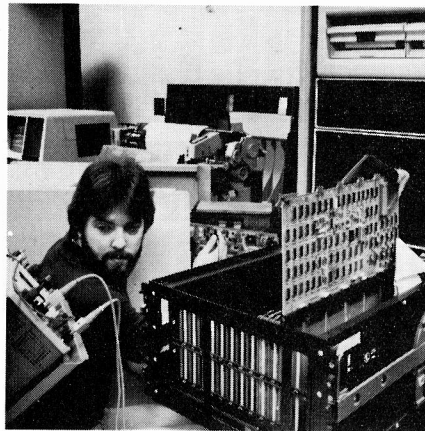
Sound interesting? Contact C. Itoh Electronics, Inc. today for more information: 5301 Beethoven Street, Los Angeles, CA 90066, (213) 306-6700. Or call ACRO Corp., exclusive representatives: Irvine, CA (714) 557-5118; Houston, TX (713) 777-1640; Cherry Hill, NJ (609) 667-4114; Chicago, IL (312) 992-2346.

 **C. ITOH
ELECTRONICS, INC.**
One World of Quality

The Single Source for Digital Products Service

DEC Module Repair, Exchange or Replacement

Photos show how we're fully equipped to repair and test DEC,* PDP-11* and TERMINAL MODULES. All modules we repair are 100% inspected.



Expert, Fast and Reliable Work

Serving the computer systems market since 1973, we are now troubleshooting, repairing and testing hundreds of modules weekly.

We can turn around a critical repair job in just 24 hours! Ten-day service is normal.

The modules we return to you are **guaranteed** to meet or exceed the manufacturer's specifications. And we prove it by submitting an inspection and test report.

For emergency service, 7 days a week, 24 hours a day, call

614/890-0939

DIGITAL PRODUCTS REPAIR CENTER

939 Eastwind Drive
Westerville, Ohio 43081
614/890-0939

*DEC and PDP are registered trademarks of Digital Equipment Corporation.

[illegible]

```
720 FOR X% = 2% TO 10% STEP 2%  
    GOTO 760 IF PEEK(WDB%+X%) = 0%  
    Z% = Z% + 1%  
    PRINT FNCS(Z%+4%,1%);  
    PRINT USING "###",6%-Z%; !PRINT LIB NUMBER  
  
730 LIB% = PEEK(WDB%+X%)  
LIB% = LIB% - 1%  
IF LIB% AND 1%  
R.NAME1% = PEEK(LIB%+2%)  
R.NAME2% = PEEK(LIB%+4%)  
PRINT FNCS(Z%+4%,20%);  
RAD$(R.NAME1%); RAD$(R.NAME2%); !RESIDENT LIBRARY NAME  
  
740 PRINT FNCS(Z%+4%,5%);  
L.PPN% = PEEK(LIB%+6%)  
L.PROJ% = SWAP%(L.PPN% AND 255%  
L.PROG% = L.FPN% AND 255%  
PRINT "[;L.PROJ%;";";L.PROG%;"]"; !PPN NUMBER  
  
750 PRINT FNCS(Z%+4%,30%);  
R.CNT% = PEEK(LIB%+26%) AND 255%  
PRINT USING "###",R.CNT%; !USER COUNT  
  
760 NEXT X%  
  
1500 IOB% = PEEK(JDBPTR%)  
SLEEP 3%  
PRINT FNCS(0,0);FNCS(0,5);FNCS(1,25);  
Job";JOB.NUMBER%;FNCS(0,7);  
FNCS(0,10);"I/O ";  
FNCS(0,9);FNCS(0,8);"Structures";  
PRINT FNCS(3,10);"OWNER SMALL PEND NEXT";  
FNCS(3,64);"FILE"  
PRINT FNCS(4,1);"CH IN UN START BUFF TRAN";  
FNCS(4,27);"BLOCK FILE NAME";  
FNCS(4,54);"RR NU CL SIZE"  
PRINT FNCS(5,1);FNCS(0,7);FNCS(0,25);  
pp pp pp ppppp ppppp ppppp";  
pppppp ppppppppppppppppppp pp";  
pp pp pp pppppp";FNCS(0,8);  
Y1% = 5%  
  
1510 FOR X% = 0% TO 30% STEP 2%  
GOTO 1590 IF PEEK(IOB%+X%) = 0%  
WCB.DDB% = PEEK(IOB%+X%)  
Y% = X%/2%  
Y1% = Y1% + 1%  
PRINT FNCS(Y1%,1%);  
PRINT USING "##",Y%;  
PRINT FNCS(Y1%,4%);  
PRINT USING "##",  
(PEEK(WCB.DDB%) AND 255%);  
GOTO 1550 IF PEEK(WCB.DDB%) AND 255%  
!START LOOP TO SEE  
! FILES, GOTO 1550 IF  
! A DDB  
  
1520 PRINT FNCS(Y1%,22%);  
PRINT USING "####",  
(PEEK(WCB.DDB%+4%) AND 255%);  
PRINT FNCS(Y1%,27%); !FIRST PART OF WCB  
  
1530 MSB.BLOCK%  
= SWAP%(SWAP%(PEEK  
(WCB.DDB%)+4%) AND 255%)  
LSB.BLOCK% = PEEK(WCB.DDB%+6%)  
BLOCKI = LSB.BLOCK%  
BLOCKI = BLOCKI + 65536  
IF BLOCKI < 0%  
LSB.BLOCK% = LSB.BLOCK%  
LSB.BLOCK% = LSB.BLOCK% + 65536  
IF LSB.BLOCK < 0%  
BLOCKI = BLOCKI + LSB.BLOCK  
PRINT USING "#####",BLOCKI; !BLOCK NUMBER  
  
1540 FCB% = PEEK(WCB.DDB%+8%) - 28%  
PRINT FNCS(Y1%,34%);"[;  
PRINT USING "###",  
(SWAP%(PEEK(FCB%+4%)) AND 255%);  
PRINT ",,";  
PRINT USING "####",  
(PEEK(PCB%+4%) AND 255%);  
PRINT "]";RAD$(PEEK(PCB%+6%));  
RAD$(PEEK(PCB%+8%));  
".,";  
RAD$(PEEK(PCB%+10%));  
PRINT FNCS(Y1%,54%);  
PRINT USING "##",  
(SWAP%(PEEK(PCB%+14%)) AND 255%);  
PRINT FNCS(Y1%,60%);PEEK(PCB%+28%);  
FILE.MSB = SWAP%(SWAP%(PEEK(PCB%+24%))  
AND 255%)  
FILE.MSB = FILE.MSB + 65536  
IF FILE.MSB < 0%  
FILE.LSB = PEEK(PCB%+26%)
```

```
FILE.LSB = FILE.LSB + 65536  
IF FILE.LSB < 0%  
PRINT FNCS(Y1%,64%);  
PRINT USING "#####",FILE.MSB+FILE.LSB  
GOTO 1590 !FINISH PRINTING FILE  
!FCB AND WCB.  
  
1550 PRINT FNCS(Y1%,7%);  
PRINT USING "##",  
{SWAP%(PEEK(WCB.DDB%+2%))  
AND 255%};  
PRINT FNCS(Y1%,10%);  
LEFT(TIMES*(PEEK(WCB.DDB%+4%))  
,5%);  
PRINT FNCS(Y1%,34%);"Device"; !PRINT DDB INFORMATION  
  
1590 NEXT X%  
PRINT FNCS(0,0);  
SLEEP 3%  
  
1600 GOTO 090  
  
20000 DEF FNCS(L9,C9)  
CURSOR CONTROL FUNCTION  
VT-100  
Arguments:  
0,0 - cursor HOME  
0,1 - cursor UP  
0,2 - cursor DOWN  
0,3 - cursor RIGHT  
0,4 - cursor LEFT  
0,5 - clear EOP  
0,6 - clear EOL  
0,7 - ANSI ON  
0,8 - ANSI OFF  
0,9 - CHARACTER ATTRIBUTES OFF  
0,10 - BOLD ON  
0,11 - UNDERSCORE ON  
0,12 - BLINK ON  
0,13 - REVERSE VIDEO ON  
0,14 - DOUBLE HEIGHT TOP 1/2  
0,15 - DOUBLE HEIGHT BOTTOM 1/2  
0,16 - SINGLE WIDTH HEIGHT  
0,17 - DOUBLE WITH SINGLE HEIGHT  
0,18 - LED OFF  
0,19 - LED 1 ON  
0,20 - LED 2 ON  
0,21 - LED 3 ON  
0,22 - LED 4 ON  
0,23 - 132 COLUMNS  
0,24 - 80 COLUMNS  
0,25 - SPECIAL GRAPHICS ON  
0,26 - SPECIAL GRAPHICS OFF  
Other arguments ,if valid, are assumed to be cursor addresses  
for direct addressing  
  
20005 L9% = INT(L9); C9%=INT(C9)  
20010 C9$ = ""  
IF L9% <> 0% THEN 20030 ELSE  
IF C9% = 0% THEN C9$ = "H" ELSE  
IF C9% = 1% THEN C9$ = "A" ELSE  
IF C9% = 2% THEN C9$ = "B" ELSE  
IF C9% = 3% THEN C9$ = "C" ELSE  
IF C9% = 4% THEN C9$ = "D" ELSE  
IF C9% = 5% THEN C9$ = "J" ELSE  
IF C9% = 6% THEN C9$ = "K" ELSE  
IF C9% = 7% THEN C9$ = "<" ELSE  
IF C9% = 8% THEN C9$ = "?21" ELSE  
IF C9% = 9% THEN C9$ = "{0m" ELSE  
IF C9% = 10% THEN C9$ = "{1m" ELSE  
IF C9% = 11% THEN C9$ = "{4m" ELSE  
IF C9% = 12% THEN C9$ = "{5m" ELSE  
IF C9% = 13% THEN C9$ = "{7m" ELSE  
IF C9% = 14% THEN C9$ = "#3" ELSE  
IF C9% = 15% THEN C9$ = "#4" ELSE  
IF C9% = 16% THEN C9$ = "#5" ELSE  
IF C9% = 17% THEN C9$ = "#6" ELSE  
IF C9% = 18% THEN C9$ = "{0c" ELSE  
IF C9% = 19% THEN C9$ = "{1c" ELSE  
IF C9% = 20% THEN C9$ = "{2c" ELSE  
IF C9% = 21% THEN C9$ = "{3c" ELSE  
IF C9% = 22% THEN C9$ = "{4c" ELSE  
IF C9% = 23% THEN C9$ = "{?3h" ELSE  
IF C9% = 24% THEN C9$ = "{?3l" ELSE  
IF C9% = 25% THEN C9$ = "(O" ELSE  
IF C9% = 26% THEN C9$ = ")O"  
20020 IF LEN(C9$) > 0 THEN 20040 ELSE 20050  
20030 IF L9% < 0% OR L9% > 24% THEN 20050  
ELSE IF C9% < 0 OR C9% > 132% THEN 20050  
ELSE C9$ = "Y" + CHR$(L9%+31%)+CHR$(C9%+31%)  
20040 FNC$ = CHR$(155%)+C9$  
GOTO 20100  
20050 FNC$ = ""  
20100 ENDC
```

```
IERROR  
EXIT
```

Word Processing*

VAX/VMS, RSTS/E,
RSX-11M

*Word-11 by
Data Processing Design, Inc.
181 W. Orangethorp Avenue
Placentia, CA 92670

On Track Systems Provides:

- Sales
- Service
- Installation
- Demonstrations
- Training
- Consulting

*At your
convenience!*

*At your
office!*

On Track Systems, Inc.

**P.O. Box 245
Ambler, PA 19002-0245
(215) 542-7133**

MAXCIM™

ON LINE INFORMATION...FROM THE PRODUCTION LINE TO THE BOTTOM LINE.

Running a manufacturing plant requires a constant flow of timely information. Not just in terms of nuts and bolts. But in dollars and cents.

And MAXCIM provides precisely that. On-line, real time information... from the production line to the bottom line.

It's the ultimate manufacturing and financial management system. Designed specifically to maximize your plant's productivity. And minimize costs. Operating on-line, in an interactive mode, MAXCIM is a comprehensive system of manufacturing and financial software modules integrated through one common database... a system which provides immediate inquiry capability to manufacturing, finance, engineering and marketing. So management can control the entire manufacturing process. While it's still in process.

Designed to run on *DEC PDP-11 and VAX minicomputers, MAXCIM is user friendly. Data entry and inquiry are menu-driven via standard CRT and teleprinter terminals. So plant personnel don't have to be data

processors to process the data they need. On the other hand, you'll like the system's sophistication. Modular flexibility. Multi-level security. Extensive audit trails. And graphics capability.

You'll also appreciate the complete system documentation we provide. And the professional training, installation and support services you receive. All backed by the experience we've gained working with hundreds of companies, large and small.

If you're looking at manufacturing and financial management systems, you can't afford to overlook MAXCIM. No other system is so comprehensive. So flexible. Or so well-supported. Call NCA today. Or write for our brochure.



NCA CORPORATION
Manufacturing/R312
388 Oakmead Parkway
Sunnyvale, CA 94086
(408) 245-7990



*DEC, PDP-11 and VAX are trademarks of Digital Equipment Corporation.

Baltimore • Boston • Chicago • Dallas • Houston • Los Angeles • New York • Portland • Sunnyvale USA / Sydney, Australia

CIRCLE 52 ON READER CARD

USRDSK

Disk Usage Summary Report for RSTS/E

By Philip Hunt, O.L.F.B.P., 6400 E. Broad Street, Columbus, Ohio 43213

USRDSK is a very valuable program designed for the system manager to see, at a glance, usage on selected or all disk devices by ascending PPN. All accounts or accounts with disk storage usage over a certain limit are reported.

Program entry

USRDSK is 'RUN'. A CCL entry point does exist, but processing is the same whether entered by CCL or 'RUN'.

After initialization, the user is requested for an output filename for the USRDSK report, the default is 'USRDSK.RPT'.

The program will then scan the device tables and system-wide logical tables and report the disks found and the number of system-wide logicals in use. The program does not have to be modified for ANY changes in disk structure, it looks up the structure when it executes. System-wide logicals are displayed on the report so the user may get a better understanding of why a certain account has a large disk usage.

The user is then requested for a search block limit for each disk in the system. The default is all usage, but any number may be entered; if it is, usage BELOW that number will not be reported. If a -1 is entered for this limit, the disk will be SKIPPED in the report.

Note: This program MUST be compiled under BP2 as it uses an RMS-INDEXED file to store account information during lookup. It is a '.TMP' file that is deleted.

Installation

BP2

OLD USRDSK
COM/OBJ BUILD/IND
TKB @USRDSK

Sample Run

```
RUN [1,3]USRDSK
USRDSK V01.03 RSTS V7.1-11 C OLFBP 11/70
Output report to <USRDSK.RPT> ?
```

Disks found:

DB0: (System)
DB1: (Private)
DB2: (Private)
DB3: (Private)

There are 53 system-wide logicals

Search block limit for DB0: < All, -1 = Skip > ? 2500
Search block limit for DB1: < All, -1 = Skip > ? 5000

Search block limit for DB2: < All, -1 = Skip > ? 5000

Search block limit for DB3: < All, -1 = Skip > ? 5000

Beginning Account lookup

Processing device — DB0:

Processing device — DB1:

Processing device — DB2:

Processing device — DB3:

Ready

Questions, or comments:

If there are any questions or comments, you may write, or call me at the below address. If you would like a copy of this program on TAPE, please send a tape and \$15.00 to the below address, specify format and density for your tape. Philip Hunt, O.L.F.B.P., 6400 E. Broad Street, Columbus, OH 43213

PPN	DB0: (System) Used/Quota	DB1: (Private) Used/Quota	DB2: (Private) Used/Quota	DB3: (Private) Used/Quota	Comments
[000,001]	13264/0	-	-	-	
[001,002]	32016/0	-	-	-	
[001,003]	8280/0	-	-	-	
[001,006]	3312/0	-	-	-	
[001,010]	7120/0	-	-	-	
[001,011]	14160/0	-	-	-	
[001,018]	8288/0	-	-	-	ST:=LB
[001,021]	2880/0	-	-	-	
[001,035]	3648/0	-	-	-	
[001,040]	-	-	-	11472/0	
[001,050]	5264/0	-	-	-	
[001,060]	-	11568/0	-	-	
[001,062]	-	10584/0	-	-	
[001,102]	7800/0	-	-	-	DB0:=DTR
[001,103]	3672/0	-	-	-	
[001,108]	4072/0	-	-	-	DB0:=C
[001,110]	2904/0	-	-	-	
[001,116]	2856/0	-	-	-	
[001,135]	7120/0	-	-	-	
[002,022]	3096/0	-	-	-	
[002,032]	3192/0	-	-	-	
[002,038]	4376/0	5688/0	-	-	
[002,042]	3272/0	-	-	-	
[002,078]	3128/0	-	-	-	
[002,080]	-	-	-	5064/0	
[002,092]	6824/0	-	-	-	
[002,098]	-	8856/0	6656/0	-	
[002,102]	-	-	-	42352/0	DB3:=PM
[002,103]	-	-	-	>65K/0	DB3:=PD
[002,107]	-	-	>65K/0	-	DB2:=CD
[002,108]	-	-	-	8032/0	DB3:=CW
[002,110]	-	21248/0	-	-	DB1:=BD
[002,111]	43208/0	-	-	-	DB0:=PH
[002,112]	-	-	-	60160/0	DB3:=CB
[002,113]	-	18432/0	-	-	DB1:=PI
[002,115]	-	14432/0	-	-	DB2:=SC
[002,117]	8192/0	-	-	-	DB0:=PM
[002,119]	-	18400/0	-	-	DB3:=CB2 DB1:=CB1 DB2:=CB
[003,022]	3088/0	-	-	-	
[003,023]	2768/0	-	-	-	
[003,028]	-	5264/0	-	-	
[003,032]	3088/0	-	-	-	
[003,036]	-	7592/0	-	-	
[003,125]	-	-	5992/0	-	DB0:=CLW0 DB3:=CLM1 DB2:=CLM2
[004,058]	-	6032/0	-	-	
[004,068]	3888/0	-	-	-	
[004,091]	-	6200/0	-	-	
[004,092]	3824/0	-	-	-	
[004,093]	-	5344/0	-	-	
[004,098]	7080/0	-	26632/0	-	
[004,101]	-	-	13824/0	-	DB2:=PMF DB1:=PP

```
1  ===== &
!
! PROGRAM: USRDSK &
! AUTHOR: PHILIP HUNT (OLFBP) &
! INITIAL DATE: 04/15/82 &
! EDIT: V01.03 &
! EDIT DATE: 05/16/82 &
! THIS PROGRAM WILL EXECUTE ON RSTS V7.0 OR LATER &
! ===== &

2  on error goto 19000 &

900  DIM DSK$(16%), dsk.typ$(16%), DSK.UNT$(16%), DSK.SKIP$(16%) &
      \ DSK%=0 &
      \ .DEFINE .MAX.DSK%=16% &
905  DIM ST$(30%), BLMT(16%) &
      I JOB STATUS &
```


YOU'RE NO SOFT TOUCH WHEN IT COMES TO SOFTWARE.



You're smart enough to buy only what you need—
and we've unbundled all of MCBA™ manufactur-
ing software so you can afford everything you want!

- GUARANTEED PERFORMANCE
- ONE-YEAR UPDATE SERVICE
- THOUSANDS OF INSTALLATIONS NATIONWIDE

In addition, INSTALLATION, TRAINING, CUS-
TOMIZATION, EXCLUSIVE SOFTLINE™ TELE-
PHONE DIAGNOSTIC SUPPORT, and other
services are available for a fee. Buy only as much
additional support as you need.



**CALIFORNIA SYSTEMS
ASSOCIATES**

2845 Mesa Verde Drive East, Suite Four,
Costa Mesa, California 92626
For demonstration call: (714) 546-9716

RSTS™ Versions Available Now!

California Systems Associates Unbundled Prices Are A Steal!

		LIST PRICE	OUR PRICE
ACCOUNTING PACKAGES:			
MCBA General Ledger	v6-1	\$2,000	\$1,350
MCBA Accounts Receivable	v6-1	2,000	1,350
MCBA Accounts Payable	v6-1	2,000	1,350
MCBA Payroll	v6-1	2,000	1,350
MCBA Fixed Assets and Depreciation	v3-0	2,000	1,350
DISTRIBUTION PACKAGES:			
MCBA Customer Order Processing	v1-1	3,000	1,850
MCBA Bill of Materials	v1-1	3,000	1,850
MCBA Inventory Management	v1-1	3,000	1,850
MCBA Purchase Order/Receiving	v1-1	3,000	1,850
MANUFACTURING PACKAGES:			
MCBA Job Costing	v1-1	2,500	1,650
MCBA Base Materials			
Requirement Plan	v1-1	2,500	1,650
MCBA Shop Floor Control	v1-1	4,000	2,750
PROFESSIONAL PACKAGE:			
MCBA Professional Time & Billing		2,000	1,600
MCBA Accountant's Client Write-up		2,000	1,600
WORDSWRIGHT wordprocessing		795	750
DATAMANAGE/REPORTWRIGHT		1,600	1,250

MCBA is a registered trademark of Mini-Computer Business Applications, Inc.
RSTS, DIBOL, DEC, PDP-11 are registered trademarks of Digital Equipment Corporation.

CIRCLE 163 ON READER CARD


```

910 DIM M$(30) &
1 I MONITOR TABLES - PART I &
1 M$(3) NULINE &
1 M$(4) MAXCNT &
1 M$(5) DEVCNT &
1 M$(7) DEVPTR &
1 M$(9) MEMBLST &
1 M$(11) JOBTBL &
1 M$(13) JBSTAT &
1 M$(15) JBWAIT &
1 M$(17) UNTCLU &
1 M$(19) UNTCNT &
1 M$(21) SATCTL &
1 M$(23) JSBTBL &
1 M$(25) SATCTM &
1100
2000
915 DIM M0$(30) &
1 I MONITOR TABLES - PART II &
1 M0$(3) FREES &
1 M0$(5) DEVNAM &
1 M0$(7) CSRTBL &
1 M0$(9) DEVOKB &
1 M0$(11) TRTCT &
1 M0$(13) JOBCNT &
1 M0$(15) RTSLSST &
1 M0$(17) ERRCTL &
1 M0$(19) SNDLST &
1 M0$(21) LOGNAM &
1 M0$(23) DEVSYN &
1 M0$(25) MEMSIZ &
2100
920 DIM READ.ACCT$(30) &
1 I USED TO HOLD ACCOUNTING DATA &
2990
999 I &
&
&
I M A I N C O D I N G A R E A &
&
&
3000
1000 ID.STG$="01.03" &
1 I SET UP VERSION/EDIT NUMBER. &
1010 DSK.SKIP$(XX$) = -1% FOR XX$=1% TO 16% &
SELECTIVE$ = "N" &
gosub 7000 Iget logical table &
PRINT "USRDSK";CHR$(9%);"V";ID.STG$;CHR$(9%); &
ERT$(0%) &
I PRINT HEADER &
1015 LINPUT "Output report to <USRDSK.RPT>";RPT.FILE$ &
rpt.file$="USRDSK.RPT" IF RPT.FILE$="" &
1016 OPEN RPT.FILE$ FOR OUTPUT AS FILE 2% &
GOTO 1020 &
1017 IERROR &
PRINT "?Open error on report file" &
goto 1015 &
1020 PRINT &
1030 CHANGE SYS(CHR$(6%)+CHR$(3%)) TO M$ &
M$(I$)=M$(I$)+SWAP$(M$(I$+1%)) FOR I$=5% TO 25% STEP 2% &
CHANGE SYS(CHR$(6%)+CHR$(12%)) TO M0$ &
M0$(I$)=M0$(I$)+SWAP$(M0$(I$+1%)) FOR I$=3% TO 25% STEP 2% &
CHANGE SYS(CHR$(6%)+CHR$(26%)) TO ST$ &
JOBNO$ = ST$(1%)/2% &
JOBNO$ = NUM1$(JOBNO$) &
JOBNO$ = "0"+JOBNO$ IF JOBNO$<10% &
TEMP.FILE$ = "TEMP"+JOBNO$+".TMP" &
I GET MONITOR TABLES , SETUP TEMP FILE NAME &
1039 I &
&
&
I P U B L I C S T R U C T U R E &
&
&
1040 UNTCNT$=M$(19%) &
SATCTL$=M$(21%) &
SATCTM$=M$(25%) &
1050 FOR DISK.TYP$=0% TO M0$(9%)-2% STEP 2% &
DEV$=CVT$(SWAP$(PEEK(M0$(5%)+DISK.TYP$))) &
FOR UNIT$=0% TO PEEK(M$(5%)+DISK.TYP$) &
J$=PEEK(UNCNT$) &
IF J$<0% OR (J$ AND 20480%)>0% THEN &
1060 &
ELSE DSK$=DSK$+1% &
GOTO 32000 IF DSK$>.MAX.DSK$ &
DSK.TYP$(DSK$) = "S" &
DSK$(DSK$)=DEV$ &
DSK.UNT$(DSK$) = UNIT$ &
1100
2100
3000
3005 ICOMPRESS THE DISK TABLE IF NEEDED &
DCNT$ = 0% &
FOR X$=1% TO DSK$ &
GOTO 3007 IF DSK.SKIP$(X$) &
DCNT$=DCNT$+1% &
SELECTIVE$ = "Y" &
GOTO 3007 IF DCNT$ = X$ &
DSK$(DCNT$) = DSK$(X$) &
BLMT(DCNT$) = BLMT(X$) &
DSK.UNT$(DCNT$) = DSK.UNT$(X$) &
DSK.TYP$(DCNT$) = DSK.TYP$(X$) &
DSK.SKIP$(DCNT$)=0% &
3007 NEXT X$ &
DSK$ = DCNT$ &
3010 for x$=1% to dsk$ &
dev$=dsk$(x$)+num1$(dsk.unt$(x$))+":" &
print "Processing device - ";dev$ &
3020 for y$=1% to 32767% &
change sys(chr$(6%)+chr$(14%)+chr$(y$)+ &
chr$(swap$(y$))+ &
string$(18%,0%)+dsk$(x$)+chr$(dsk.unt$(x$))+ &
chr$(1%)+string$(4%,0%)) to st$ &
ppn$=fnpad$(st$(8%),3%)+fnpad$(st$(7%),3%) &
blk = st$(5%)+(256.0*st$(6%)) &
qta = st$(27%)+(256.0*st$(28%)) &
GOTO 3060 IF BLK<BLMT(X$) &
3030 writ$ = 0% &
get #1$, key#0% eq ppn$ &
goto 3050 &
3040 lrec not found here &
blocks(dd$)=-1% for dd$=1% to 16% &
quota(dd$)=-1% for dd$=1% to 16% &
writ$ = -1% &
3050 blocks(x$) = blk &
quota(x$) = qta &
update #1$ unless writ$ &
put #1$ if writ$ &
3060 next y$ &
print "?Too many accounts on ";DEV$;" - ABORTING" &
goto 32767 &
3100 next x$
3110 ltable is now loaded, lets print it &
lent$=0% &
gosub 5000 &
find #1$, key#0% ge "000000" &
3120 get #1$ &
3125 print #2$,"["+left(ppn$,3%)+", "+right(ppn$,4%)+"]";tab(10%); &
proj$ = val(left(ppn$,3%)) &
prog$ = val(right(ppn$,4%)) &
3130 for dd$=1% to dsk$ &
GOTO 3131 IF DSK.SKIP$(X$) &
blocks$ = fnspace$(blocks(dd$),5%) &
quota$ = num1$(quota(dd$)) &
quota$ = quota$+space$(5%-len(quota$)) &
blocks$=">=65K" if blocks(dd$)=65535. &
outst$ = " " if blocks(dd$)=-1% &
print #2$, tab(15%+(14%*(dd$-1%)));outst$; &
3131 next dd$ &
print #2$, " "; &
for xx$=1% to lgcnt$ &

```


PDP-11, VAX-11, LSI-11 users...

don't buy a file transfer software package unless you know the answers to these important questions...

Q Hardware Requirements

Does the package require any special communications hardware?

A No. XOREN IPL-11 uses a link between two standard asynchronous terminal ports.

...

Q Local and Remote Links

Can the package link computers on either the same site or different sites?

A Yes. For a local link a direct line may be used; for a remote link, XOREN IPL-11 uses a telephone line and modems or acoustic couplers.

...

Q Error Checking

Does the package include full communications error checking?

A Yes – and XOREN IPL-11 recovers from errors by re-transmitting *only* the blocks affected.

...

Q Operating Systems Supported

What operating systems are supported by the package?

A Versions of XOREN IPL-11 are available for most DEC operating systems, including RSX-11M, RSX-11M PLUS, RT-11, CTS-300, RSTS/E, CTS-500, VAX/VMS, P/OS, TSX PLUS.

...

Q Linking Different Operating Systems

Is it possible to link different operating systems and transfer files between them?

A Yes, all versions of XOREN IPL-11 are compatible, so that, for example, files can be transferred from a RSTS/E system to an RSX-11M system.

...

Q Software Licence Requirements

Is the package written in a high-level language, for which the user must have a licence?

A No, all versions of XOREN IPL-11 are written in assembly language, thereby maximising processing efficiency.

...

Q Types of File Handled

Can the package transfer binary files (object and task files) as well as source files?

A Yes.

...

Q Program Size

How much memory does the package use?

A Memory requirements for XOREN IPL-11 are approximately 15K words for RT-11, CTS-300 and TSX PLUS versions, 18K for RSTS/E and CTS-500 versions, and 21K for RSX-11M, RSX-11M PLUS, VAX/VMS versions. (For RSX-11M and RSX-11M PLUS a 15K 'compact' version is also available).

Q Control of Transfers

Can file transfers in either direction be controlled and monitored from either computer?

A Yes.

...

Q Multi-Channel Interface Cards

Can the package use for the link a port on a DZ11 or DH11 interface card, as an alternative to a single port device, such as a DL11?

A Yes: Except for the RT-11, CTS-300 and TSX PLUS versions, XOREN IPL-11 may use any port supported by the operating system's standard terminal driver. (Under TSX PLUS, only a DL11 or equivalent may be used. Under RT-11/CTS-300, a DZ11 may be used instead, but other users will be affected).

...

Q Transparency to other users

When the package operates under multi-user operating systems, can other jobs continue running unaffected?

A Yes.

...

Q Automatic Modes of Operation

In addition to normal interactive operation, can the package operate in batch mode or under the control of an indirect command file?

A Yes, all versions of XOREN IPL-11 have a facility of this kind.

...

Q Remote Activation

Can a link be established without an operator at the remote site?

A Yes, provided that the remote computer is running a multi-user operating system.

...

Q File attributes

Are file attributes handled correctly?

A Yes, XOREN IPL-11 transfers file attributes with the file wherever applicable.

...

Q File Formats

When files are transferred between operating systems which use different file formats, are facilities provided for the necessary format conversions?

A Yes. From this point of view, the supported operating systems fall into two groups. The FILES-11 group are RSX-11M, RSX-11M PLUS, and VAX/VMS, while RT-11, CTS-300, TSX PLUS, RSTS/E and CTS-500 form the RT-11 group. Format conversion is only necessary when files are transferred between these two groups. It may be performed using a standard DEC utility such as FLX, or using XOREN FLC-11, a special-purpose utility which has the advantage that it can handle RT-11 files on FILES-11 volumes.

Q File Transfer Rates

How long do file transfers take?

A This depends on such factors as the CPU model, operating system, and line speed. Typical transfer rates for XOREN IPL-11 (based on 512-byte blocks) are 40 to 60 blocks per minute at a line speed of 9600 baud (but 20 to 25 for RSTS/E and TSX PLUS), and 10 to 12 blocks per minute at 1200 baud.

...

Q CPU Use

How much CPU time does the package use?

A Tests on a PDP-11/34 show that XOREN IPL-11 uses less than 2% of CPU time at a line speed of 1200 baud, and less than 30% at 9600 baud.

...

Q Future Support

Is provision made for the supply of future enhancements, and new releases of the package to run under new versions of operating systems?

A Yes.

...

XOREN IPL-11 is a low-cost, easy to implement package for transferring files between two DEC PDP-11, LSI-11 or VAX-11 computers even when they have different operating systems.

A licensed copy of the package run at each computer enables files to be transmitted over direct or telephone links using standard asynchronous terminal ports.

XOREN IPL-11

XOREN IPL-11, the simple solution to DEC file transfers.

UK Office

Xoren Computing Limited
28 Maddox Street
London W1R 9PF
Telephone (01) 629 5932

USA Distributors

Softpak Associates
634½ Venice Boulevard
Marina Del Rey
California 90291
Telephone (213) 822 1830

USA Office

Xoren Computing
Suite 800
50 Staniford Street
Boston MA 02114
Telephone (617) 720 1096

Midcom Corporation
1940 N Tustin, Suite 117
Orange
California 92665
Telephone (714) 998 6041



CIRCLE 100 ON READER CARD

Dreaming of

Product Name: Dreams Version 5.0

Since its first sale in 1979 Dreams has grown in capability and user acceptance. It is now in use on over 40 RSTS/E systems around the country.

Special Features:

- DECnet compatibility—message transmission to distant nodes. Includes message queuing to unavailable nodes.
- Invoke your favorite style of editing (EDT, DECword, WORD-11, TECO, etc.) with a smooth transition to and from the editor.
- Flexible method for accessing and maintaining multiple mail files.
- Subjects for mail files as well as individual messages.
- Retract unread messages.
- Recover your last deleted message.
- Specify times as well as dates in relative or absolute form to control message appearance or expiration or to narrow selection criteria.
- Full compatibility with Batch. This opens up a world of possibilities for keeping abreast of unattended operations and for implementation of a repetitive reminder system based on day of the week or other longer intervals.
- System manager may assign defaults for accounts, projects, and the entire system including the assignment of certain privileges.

The Dreams package consists of over 40,000 lines of source code in more than 70 modules plus significant documentation both as documents and as on-line help. CSPCOM or BASIC-Plus-2 builds these sources into only 5 Dreams tasks: TELL, MAIL, WHO, SMASH, and MAN-AGE (plus POSTMN for the DECnet version). Computers with sufficient memory may use the customized resident library and resident run-time system.

A VAX version will be available later.

Typical Electronic Mail Features are also included in Dreams:

- Send to names, nicknames, or groups.
- Scan, reply, forward, or store for later appearance.
- New, old, priority, or suppressed messages for each mail file.
- Automatic routing of messages.
- Many other convenient features.

Ordering information:

Available on 9 track 800 or 1600 BPI tape. Multiple CPU discount schedule:

First License	\$3000.00
Dreams/5	
DECnet modules	\$3000.00
Second and Third License	40% Discount
Fourth and Fifth License	50% Discount
Sixth and up	70% Discount
Educational Institutions	Additional 50% off the total

Maintenance and new releases: Annual fee of 12% of current list price after the first year.

For more information contact:

**Tom Burnett
DCXX Software Services
Dickinson College
Computer Center
Carlisle, PA 17013
717-245-1513**

RSTS/E, VAX, DECnet, and DECword are trademarks of Digital Equipment Corporation.
WORD-11 is a trademark of Data Processing Design, Inc.

```

\      if proj%=logs.proj$(xx%) &
\      and prog%=logs.prog$(xx%) then &
\      print #2%, logs.name$(xx%);" "; &
3135  next xx% &
\      print #2% &
\      lent% = lent%+1% &
\      if lent% > 50% then gosub 5000 &
3140  goto 3120 &
4000  lend of temp file, lets finish up &
\      print #2%, chr$(12%); &
\      close #1%, #2% &
\      goto 32767 &
5000  leop routine &
\      page%=page%+1% &
\      lent%=0% &
\      TPOS% = 5% &
\      SEL% = 50% IF SELECTIVE%="Y" &
\      SEL% = "Selective " IF SELECTIVE%="Y" &
\      print #2%, chr$(12%);"Date: ";date$(0%);" ";TIME$(0%);tab(TPOS%); &
\      SEL%;"Disk Management Report";tab(108);"USRDSK-"; &
\      "Page: ";PAGE% &
\      PRINT #2%, TAB(53%);ERT$(0%) &
\      print #2% &
\      print #2%, "----PPN---- "; &
\      for t%=1% to dsk% &
\      DV$=dsk$(t%)+num1$(dsk.unt$(t%))+":" &
\      print #2%, tab(18%+((t%-1%)*14%));DV$; &
\      next t% &
\      print #2% &
\      for t%=1% to dsk% &
\      typ$="(Unknown)" &
\      typ$="(System)" if dsk.typ$(t%)="S" &
\      typ$="(Private)" if dsk.typ$(t%)="P" &
\      print #2%, tab(16%+((t%-1%)*14%));typ$; &
\      next t% &
\      print #2% &
\      for t%=1% to dsk% &
\      print #2%, tab(17%+((t%-1%)*14%)); &
\      LM% = NUM1$(BLMT(T%)) &
\      LM% = LEFT("00000",5%-LEN(LM%))+LM% &
\      LM% = "ALL" IF BLMT(T%)=0% &
\      PRINT #2%, ">";LM%; &
\      NEXT T% &
\      print #2% &
\      for t%=1% to dsk% &
\      print #2%, tab(14%+((t%-1%)*14%));" Used/Quota"; &
\      NEXT T% &
\      PRINT #2%, " Comments"; &
\      print #2%, CR+LF &
5100  RETURN &
7000  isystem logical getter &
\      DIM MM$(30%),LOGS.DEV$(60%),LOGS.PROJ$(60%),LOGS.PROG$(60%)
\      DIM LOGS.NAME$(60%)
7005  CHANGE SYS$(CHR$(6%)+CHR$(-12%)) TO MM% &
7010  \ LGCNT% = 0% &
\      D$=MM$(21%)+SWAP$(MM$(22%)) &
\      WHILE PEEK(D%)<-1% &
\      DO$=PEEK(D%) &
\      GOTO 7020 IF DO$=0% &
\      Q$=CVT$(SWAP$(PEEK(D%+4%))) &
\      Q$=PEEK(D%+6%) &
\      Q$=Q$+NUM1$(Q% AND 255%) UNLESS Q%=0% &
\      Q$=Q$+" " &
\      LGCNT%=(LGCNT%+1% &
\      LOGS.DEV$(LGCNT%)=Q$ &
\      Q$=PEEK(D%+8%) &
\      LOGS.PROJ$(LGCNT%)=SWAP$(Q%) AND 255% &
\      LOGS.PROG$(LGCNT%)=Q% AND 255% &
\      Q$=Q$+"="+RAD$(DO$)+RAD$(PEEK(D%+2%)) &
\      LOGS.NAME$(LGCNT%)=Q$ &
\      D$=D$+10%
7020  NEXT &
7030  return &
19000  ! &
\      &
\      ! ERROR HANDLING &
\      &
\      &
19001  resume 3040 if erl=3030 and err=155% &
\      resume 3100 if erl=3020 and err=5% &
\      resume 32767 if erl=1015 and err=11 &
\      resume 32767 if erl=2100 &
\      resume 1017 if erl=1016 &
\      resume 3000 if erl=2990 and err=5% &
\      resume 4000 if erl=3120 and (err=11% or err=155%) &
19900  print "zurdsrk - fatal error #";err;" (";ERT$(err);") at ";ERL &
\      goto 32767 &
20000  def fnpad$(arg$,lg%) &
\      pdl% = num1$(arg%) &
\      pdl% = len(pdl%) &
\      pdl% = left("000",lg%-pdl%)+pdl% &
\      fnpad% = pdl% \ fndend &
20100  def fnspc$(arg$,lg%) &
\      pdl% = num1$(arg%) &
\      pdl% = len(pdl%) &
\      pdl% = left(" ",lg%-pdl%)+pdl% &
\      fnspc% = pdl% \ fndend &
20200  def fnspe$(arg$,lg%) &
\      pdl% = num1$(arg%) &
\      pdl% = len(pdl%) &
\      pdl% = pdl%+left(" ",lg%-pdl%) &
\      fnspe% = pdl% \ fndend &
32000  PRINT "?Too many disks - aborting" \ GOTO 32767
32767  END

```

BRAEGEN MPD HAS THE DISK STORAGE SUBSYSTEM YOU NEED

INTRODUCING THE DSL-SERIES OF DISK STORAGE SUBSYSTEMS

Braegen's Minicomputer
Peripherals Division is
committed to providing
the systems and
peripherals
DEC-users need.



BENEFITS

- Compatible with *DEC *PDP-11 and *VAX Massbus systems.
- Provides *RM03/RM05 Emulation with enhancements.
- Supports mixed drive sizes.
- Supports DEC Dual Port Diagnostics.
- Supports Overlap Seeks.
- Fast delivery.
- Local service and quantity discounts available.

For more information on the DSL-Series of mass storage subsystems or any of our full line of DEC-compatible computer systems, disk and tape subsystems, printers, memories, multiplexers and more, please call or write:

Marketing Dept:



**BRAEGEN
MPD**

3320 East La Palma Avenue
Anaheim, California 92806
Telephone (714) 520-9200

"Be sure to ask about our field service force in over 50 cities nationwide."

*Trademarks of Digital Equipment Corporation

David Spencer, Infinity Software Corporation

Some EDT Wishes Granted

Last issue I noted that there is a rather limiting problem with the EDT implementation for PDP-11s. This problem is an outgrowth of the difference between VAX (where EDT grew up), a machine with nearly unlimited address space, and the PDP-11 with a limit of 64K bytes of addressing space.

Not one to let a problem just sit, I invested part of my Labor Day weekend and looked into the problem.

1.0 THE PROBLEM

I disassembled the modules that request additional memory for storing new key definitions and buffer names. As noted before, it is precisely true that there is a fixed block of memory that simply gets appended to. When EDT runs out, that's it. (Apparently VAX must just keep asking for memory and expanding dynamically.)

2.0 THE SOLUTION

I don't have the perfect solution, which would have been to write a garbage collect routine on the work block. That's a little out of the scope of what I can handle. What I have done, though, is locate the work block and make it larger.

The following object module can be entered by hand (it's short, really) and used to create an EDT task with more work space. Notice the comment where I note the old values for the work space and use the new . . .

```
.TITLE      IODATA      , < Disassembled EDT data block > , 07, DEC/ DJ5, 05-Sep-82

.IDENT      /02.01A/

.PSECT      IODATA

IO.FNF      = = 5
IO.BUF::BLKB      400
IO.STS::WORD      0
IO.RFM::WORD      0
SY.AUD::
SY.COM::WORD      4
      .BLKB      1020
SY.PRI::WORD      6
      .BLKB      1020
SY.ALT::WORD      10
      .BLKB      1020
IO.RHB::BLKB      4
IO.WF::WORD      0
IO.LPR::BYTE      0
IO.PRT::BYTE      0
WF.SIZ::WORD      0
WF.IFI::WORD      0
IONAME::WORD      0
IONLEN::WORD      0
JOUNAM::BLKB      40
S.BUGC::ASCII      /BUG CHECK / ; Those are three spaces . . .
S.HEAP::BLKW      4230 ; Stock value is 1750
S.HPOI::WORD      4230 ; Stock value is 1750
S.WORK::WORD      0
WC.TIM::WORD      0
WC.BUK::BLKB      20
WC.REF::BLKB      20
WC.DIR::BLKB      10
```

```
WC.CAC::BLKB      10000
G.MCR ::WORD      24577
      .BLKB      120
G.LEN ::WORD      0
```

.END

In my version of the IODATA module, the work block (called S.HEAP) has been increased from 1000 (base ten) words to 2200 (base ten) words. This increases the size of all versions of EDT, flat, overlaid, or shared, by 1K word. In other words, flat EDT becomes 28K, overlaid 20K, and shared still has a 21K resident library, but the task increases to 8K.

Please note that I have not solved the problem of running out of work space; I have just made it harder for the problem to happen.

To create a new copy of EDT using IODATA.MAC, use the following general steps:

1. Create IODATA.MAC from the above code with your favorite editor.
2. Assemble IODATA with \$MAC.TSK like this:

```
RUN $MAC
MAC>IODATA=IODATA
MAC>IZ
```
3. Verify the module was entered correctly with a checksum from PAT.TSK.

```
RUN $PAT
PAT>NL:=IODATA/CS:056067,IODATA
PAT>IZ
```

If the checksum doesn't match then go back and review your entry of the IODATA module.
4. Get out your RSTS distribution and restore EDT??? to the current account. On the 7.1 distribution it is on the second of the three tape set of CUSPs.
5. Replace the IODATA module in the EDTLIB library:

```
RUN $LBR
LBR>EDTLIB/RP=IODATA
MODULE "IODATA" REPLACED
```

LBR>IZ

6. There are three choices of command files for building EDT. They are:
 EDTBLD for a flat-out task
 EDTOVb for an overlaid task
 EDTSHR for a task with a resident library
 Task build with the configuration type you desire (my favorite is shared, used as the example here):

```
RUN $TKB
TKB>@EDTSHR
```
7. Add the symbol table to allow symbolic patching (and to create the resident library if selected).

```
RUN $MAKSIL
MAKSIL V7.1-11 RSTS/E V7.1-11 Timesharing
Resident Library name? EDT
Task-built Resident Library input file <EDT.TSK>? EDT.TSK
Include symbol table (Yes/No) <Yes>? YES
Symbol table input file <EDT.STB>? EDT.STB
Task Image SIL output file <EDT.SIL>? EDT.SIL
EDT built in 8 K-words, 460 symbols in the directory
EDT.SIL renamed to EDT.SIL<104>
```


8. Copy the task to the system account with a protection of "<104>". If the shared version was selected, copy the resident library to the account required and name it with a protection code of "<42>" and add it at the right address with UTILITY.

Before making the final transfers of the new copy of EDT, save the old one for a while until you are absolutely sure that the IODATA module was entered correctly.

If the previous instructions were confusing or too general for you, please don't attempt to perform them. Get somebody who can. I am not trying to insult anybody's intelligence. It's just that performing the installation wrong can screw things up for a while and is hard to live down afterwards.

3.0 ANOTHER WISH GRANTED

A number of issues ago I listed my EDT wishes. One of the things I wanted was the journal file created on the current account. Currently EDT creates the journal file on the disk and account of the input file to be edited. The following patch file, called "JOURNAL.PATCH", always creates the journal file on the current account on the public structure, otherwise known as "SY:".

This patch is for the EDT released with RSTS V7.1, and beyond. It will not work for the version 7.0 EDT release, which is different. If you are still running RSTS 7.0, there is no problem with installing and running the RSTS 7.1 release of EDT.

.TITLE IOMOD ,< Patch for journal file>,07,DEC/DJS,05-Sep-82

.IDENT /02.02B/

```
.GLOBL    BL$MOV
.GLOBL    SY.AUD
```

PSECT IOCOD

\$\$\$ =

```

      . = $$$$. + 000210
      CALL      TSTJOU

```

. = \$\$\$\$. + 000566
CALL TSTJOU

. = \$\$\$\$ + 003242
CALL J.COPY

```

TSTJOU:  MOV#B      (R2),406
          CMP#B      @#406,S#Y.AUD
          BNE        ITS.OK

```

```

FQ.CLR: CMP      @#432,@"NL
        BEQ      ITS.OK
        CLR      @#410
        CLR      @#432
        CLR      @#434

```

ITS.OK: RETURN

```
J.COPY: CALL    FQ.CLR
          JMP     BL$MOV
```

.END

To install this patch, perform these general procedures:

1. Create JOURNAL.MAC from the listing above with your favorite editor.
2. Extract the IOMOD module to be patched from the EDT object library.
RUN \$LBR

RSTS ARCADE

Infinity Software has arcade quality games for VT52 and VT100 terminals.

BLKADE Up to a total of eight players try to fence in their opponents in this game of skill and strategy.

PACKER Collect the radio-active dust from the maze before the four robots catch you. (VT100 only.)

INVADE Protect the planet with your laser base from the alien invaders from space. (VT100 only.)

SUBS Locate and sink your opponent's submarine before he sinks you.

STRWRS You have five minutes to destroy the Death Star before it destroys your moon base.

WIGGIT Drop water crystals in front of the Martian Wiggits before they clone and capture you! (VT100 only.)

Games are \$39.95 each and come complete with User Guides. Distribution is made on 800 or 1600 BPI magtape (other media arranged at higher cost). California residents please add 6½ percent.

GIVE A GAME FOR CHRISTMAS!

For ordering information, please write or call:

INFINITY SOFTWARE CORPORATION

2210 Wilshire Blvd, Suite 801
Santa Monica, California 90403
(213) 820-2702

CIRCLE 87 ON READER CARD

LBR > IOMOD = EDTLIB/EX:IOMOD

 $LBR > \uparrow Z$

3. Next, assemble the patch file.

RUN \$MAC.TSK

MAC>JOU PAT=JOU PAT

MAC > 12

4. Then merge the two modules to perform the patch.

```

RUN $PAT.TSK

```

PAT>NEWMOD=IOMOD/CS:001507,JOUPAT/CS:023007

PAT > 1Z

- If you got a checksum error, stop here. Either there was a patch to the IOMOD module previously (unlikely unless you did it yourself), or the JOUPAT source was entered incorrectly. Go back and review your typing and try again until there are no errors. If all is well, replace the corrected module back into the EDT object library.

Dennis DeMaria, Watertown, Mass.



Total Office Automation

for DEC VMS, RSTS/E, RSX-11M, 11M+, TSX+, RT11

CT*OS Word Processing

- Full editing capability — global search/replace, cut/paste, headers, footers, auto numbering, boiler-plate libraries, document assembly, auto bold, underline, uppercase. Movement by word, paragraph, sentence, page, etc. Status line indicates page, line, column position in addition to the mode, document number, and account.
- Simultaneous print and edit — variety of serial letter quality and draft printers. Full-page CRT compatibility.
- Full production word processing with multiple terminals and multiple printers. CPU efficient and easy to learn. Interfaces with DP files.
- Adaptable to your own application with user-defined keys and list processing.

Exectronic Mail™

- 60,000+ users — 30kb for first user, each additional user 10kb, one-keystroke instructions.
- Management communications system — directory of users, log of communications, acknowledge, and nudge capability, and more. Transfer any file on the system.
- Create and read your message with a full-featured word processor.

Electronic Spreadsheet

- Budgets and cash flow projections — create with the spreadsheet package, edit in CT*OS, and send with Exectronic Mail™ to the head office. Or print to the printer or to a disk file for inclusion in other reports. Or send to the typesetter.

Spelling Correction (more than a dictionary)

- Global correction directly on the word processing document, while user dictionary is updated with special terminology — 80,000 word dictionary.

East Coast

Dennis A. Wills
Compu-Tome, Inc.
(213) 796-9371

Midwest

Paul Harris
Gina Harris
Mountain W. Software
(213) 796-9374

Pacific

Diane Sullivan
Ted Dresser
Discom-
(805) 499-6256
(213) 796-9375

COMPU-TOME, INC.

234 East Colorado Boulevard
Pasadena, CA 91101

FORMS

By Jeff Corbett, Programmer/Operator, Oconee County, Walhalla, SC 29691

1.0 Introduction

FORMS is a program designed to make it easier to maintain the proper settings for a line printer spooling job. When the time came to begin setting up forms for the various types of paper that would find itself in the lineprinter, I realized that remembering all of the form names, page lengths and width settings that I would be using could be quite a headache, and neither PLEASE, QUE nor SPOOL had any intention of remembering for me. Thus FORMS was born.

2.0 FORMS User's Guide

FORMS first major task upon start up consists of looking through the message receiver table, making sure the receiver OPSER exists, and storing every occurrence of a "LP?SPL" receiver. If only one is found it is assumed to be the only possible target for form changing and FORMS proceeds to the next operation; otherwise it displays a menu listing of all the spoolers it found and asks for the user's choice. The choice may be specified by number or by receiver ID.

Once the correct spooler has been decided upon, FORMS will read FORMS.DAT and display a menu listing all of the form definitions that it has found. It will then ask for the user's choice. The desired form may be indicated either by the form number or the form name. Typing "0" (zero) will cause the spooler to print the current form and switches, or will align the current form in conjunction with "/A" (see next paragraph). The length of the form name given is all that is checked against the table of legal forms, therefore typing "NO" will match the first form in the table beginning with "NO" (probably NORMAL).

The only switch that FORMS currently has available is the "/A" switch. The "/A" option causes the "/ALIGN" switch to be appended to the form selected, if not already present. This causes the spooler to print a test alignment pattern and to wait to be sent confirmation before resuming operations.

After the form has been selected, FORMS will send OPSER the following command line:

```
INT <spooler> /FOR <formname> / <switches>
```

and then exit, unless the "/ALIGN" switch is included in the <switches> (see next paragraph). Since OPSER has a say in who sends to the spooler, if the user is not one of the OPSER valid operators all this work will be for naught. However, if the user is on OPSER's table, the request will be forwarded to the spooler which will then process the form definition and send the Operator Services Console the confirmation of the form change. FORMS exits with the "Exit with No Prompt Message" SYS call because the process isn't quite done until the OSC receives the confirmation, and it is assumed that FORMS will be run at the OSC to receive the messages from the spooler.

If the "/ALIGN" switch was in the form definition or requested by the user, FORMS will print "Standing By..." and go into an input wait. The user should wait for the spooler to go through the forms alignment process and then type the appropriate response (either "RETRY" for another check or "GO" to tell the spooler to go ahead). FORMS will repeat this step until a "GO" is typed, after which it will exit.

The FORMS CCL command accepts the following format:

```
FORMS/ <form> [/A] <spooler>
```

Note: The <form> and <spooler> may be identified by either name or by number here as in their respective menus. The <spooler> is not necessary and will be ignored if only one spooler is present. The "/A" switch was explained earlier. If a form or spooler is found to be incorrect, the appropriate menu will be printed and the user will be prompted.

3.0 Installation Notes

Before FORMS can be used, it must be compiled, protected <232>, and the data file "FORMS.DAT" must be placed in the same account. This data file should contain all of the forms used at your installation. A sample data file is included at the end of this article. The CCL definition:

```
CCL FORM-S=[PPN]FORMS.BAC;PRIV 30000
```

is supported by FORMS. The <232> protection code and the PRIV option are only necessary if there are non-privileged OPSER "operators" that will be using FORMS, otherwise a <124> will suffice.

FORMS may be compiled by CSPCOM if desired, but comment line 120 must be removed before compiling or CSPCOM will generate Err 0, and quit compiling. FORMS was developed under RSTS V7.0-07, and minor changes may be required under newer releases of RSTS. I hope you will find this program of use and that it will keep you in top form.

```

1      EXTEND
21     &
3      &
4      &
5      &
6      &
7      &
8      &
9      &
10     &
11     &
12     &
13     &
14     &
15     &
16     &
17     &
18     &
19     &
20     &
21     &
22     &
23     &
24     &
25     &
26     &
27     &
28     &
29     &
30     &
31     &
32     &
33     &
34     &
35     &
36     &
37     &
38     &
39     &
40     &
41     &
42     &
43     &
44     &
45     &
46     &
47     &
48     &
49     &
50     &
51     &
52     &
53     &
54     &
55     &
56     &
57     &
58     &
59     &
60     &
61     &
62     &
63     &
64     &
65     &
66     &
67     &
68     &
69     &
70     &
71     &
72     &
73     &
74     &
75     &
76     &
77     &
78     &
79     &
80     &
81     &
82     &
83     &
84     &
85     &
86     &
87     &
88     &
89     &
90     &
91     &
92     &
93     &
94     &
95     &
96     &
97     &
98     &
99     &
100    &
101    &
102    &
103    &
104    &
105    &
106    &
107    &
108    &
109    &
110    &
111    &
112    &
113    &
114    &
115    &
116    &
117    &
118    &
119    &
120    &
121    &
122    &
123    &
124    &
125    &
126    &
127    &
128    &
129    &
130    &
131    &
132    &
133    &
134    &
135    &
136    &
137    &
138    &
139    &
140    &
141    &
142    &
143    &
144    &
145    &
146    &
147    &
148    &
149    &
150    &
151    &
152    &
153    &
154    &
155    &
156    &
157    &
158    &
159    &
160    &
161    &
162    &
163    &
164    &
165    &
166    &
167    &
168    &
169    &
170    &
171    &
172    &
173    &
174    &
175    &
176    &
177    &
178    &
179    &
180    &
181    &
182    &
183    &
184    &
185    &
186    &
187    &
188    &
189    &
190    &
191    &
192    &
193    &
194    &
195    &
196    &
197    &
198    &
199    &
200    &
201    &
202    &
203    &
204    &
205    &
206    &
207    &
208    &
209    &
210    &
211    &
212    &
213    &
214    &
215    &
216    &
217    &
218    &
219    &
220    &
221    &
222    &
223    &
224    &
225    &
226    &
227    &
228    &
229    &
230    &
231    &
232    &
233    &
234    &
235    &
236    &
237    &
238    &
239    &
240    &
241    &
242    &
243    &
244    &
245    &
246    &
247    &
248    &
249    &
250    &
251    &
252    &
253    &
254    &
255    &
256    &
257    &
258    &
259    &
260    &
261    &
262    &
263    &
264    &
265    &
266    &
267    &
268    &
269    &
270    &
271    &
272    &
273    &
274    &
275    &
276    &
277    &
278    &
279    &
280    &
281    &
282    &
283    &
284    &
285    &
286    &
287    &
288    &
289    &
290    &
291    &
292    &
293    &
294    &
295    &
296    &
297    &
298    &
299    &
300    &
301    &
302    &
303    &
304    &
305    &
306    &
307    &
308    &
309    &
310    &
311    &
312    &
313    &
314    &
315    &
316    &
317    &
318    &
319    &
320    &
321    &
322    &
323    &
324    &
325    &
326    &
327    &
328    &
329    &
330    &
331    &
332    &
333    &
334    &
335    &
336    &
337    &
338    &
339    &
340    &
341    &
342    &
343    &
344    &
345    &
346    &
347    &
348    &
349    &
350    &
351    &
352    &
353    &
354    &
355    &
356    &
357    &
358    &
359    &
360    &
361    &
362    &
363    &
364    &
365    &
366    &
367    &
368    &
369    &
370    &
371    &
372    &
373    &
374    &
375    &
376    &
377    &
378    &
379    &
380    &
381    &
382    &
383    &
384    &
385    &
386    &
387    &
388    &
389    &
390    &
391    &
392    &
393    &
394    &
395    &
396    &
397    &
398    &
399    &
400    &
401    &
402    &
403    &
404    &
405    &
406    &
407    &
408    &
409    &
410    &
411    &
412    &
413    &
414    &
415    &
416    &
417    &
418    &
419    &
420    &
421    &
422    &
423    &
424    &
425    &
426    &
427    &
428    &
429    &
430    &
431    &
432    &
433    &
434    &
435    &
436    &
437    &
438    &
439    &
440    &
441    &
442    &
443    &
444    &
445    &
446    &
447    &
448    &
449    &
450    &
451    &
452    &
453    &
454    &
455    &
456    &
457    &
458    &
459    &
460    &
461    &
462    &
463    &
464    &
465    &
466    &
467    &
468    &
469    &
470    &
471    &
472    &
473    &
474    &
475    &
476    &
477    &
478    &
479    &
480    &
481    &
482    &
483    &
484    &
485    &
486    &
487    &
488    &
489    &
490    &
491    &
492    &
493    &
494    &
495    &
496    &
497    &
498    &
499    &
500    &
501    &
502    &
503    &
504    &
505    &
506    &
507    &
508    &
509    &
510    &
511    &
512    &
513    &
514    &
515    &
516    &
517    &
518    &
519    &
520    &
521    &
522    &
523    &
524    &
525    &
526    &
527    &
528    &
529    &
530    &
531    &
532    &
533    &
534    &
535    &
536    &
537    &
538    &
539    &
540    &
541    &
542    &
543    &
544    &
545    &
546    &
547    &
548    &
549    &
550    &
551    &
552    &
553    &
554    &
555    &
556    &
557    &
558    &
559    &
560    &
561    &
562    &
563    &
564    &
565    &
566    &
567    &
568    &
569    &
570    &
571    &
572    &
573    &
574    &
575    &
576    &
577    &
578    &
579    &
580    &
581    &
582    &
583    &
584    &
585    &
586    &
587    &
588    &
589    &
590    &
591    &
592    &
593    &
594    &
595    &
596    &
597    &
598    &
599    &
600    &
601    &
602    &
603    &
604    &
605    &
606    &
607    &
608    &
609    &
610    &
611    &
612    &
613    &
614    &
615    &
616    &
617    &
618    &
619    &
620    &
621    &
622    &
623    &
624    &
625    &
626    &
627    &
628    &
629    &
630    &
631    &
632    &
633    &
634    &
635    &
636    &
637    &
638    &
639    &
640    &
641    &
642    &
643    &
644    &
645    &
646    &
647    &
648    &
649    &
650    &
651    &
652    &
653    &
654    &
655    &
656    &
657    &
658    &
659    &
660    &
661    &
662    &
663    &
664    &
665    &
666    &
667    &
668    &
669    &
670    &
671    &
672    &
673    &
674    &
675    &
676    &
677    &
678    &
679    &
680    &
681    &
682    &
683    &
684    &
685    &
686    &
687    &
688    &
689    &
690    &
691    &
692    &
693    &
694    &
695    &
696    &
697    &
698    &
699    &
700    &
701    &
702    &
703    &
704    &
705    &
706    &
707    &
708    &
709    &
710    &
711    &
712    &
713    &
714    &
715    &
716    &
717    &
718    &
719    &
720    &
721    &
722    &
723    &
724    &
725    &
726    &
727    &
728    &
729    &
730    &
731    &
732    &
733    &
734    &
735    &
736    &
737    &
738    &
739    &
740    &
741    &
742    &
743    &
744    &
745    &
746    &
747    &
748    &
749    &
750    &
751    &
752    &
753    &
754    &
755    &
756    &
757    &
758    &
759    &
760    &
761    &
762    &
763    &
764    &
765    &
766    &
767    &
768    &
769    &
770    &
771    &
772    &
773    &
774    &
775    &
776    &
777    &
778    &
779    &
780    &
781    &
782    &
783    &
784    &
785    &
786    &
787    &
788    &
789    &
790    &
791    &
792    &
793    &
794    &
795    &
796    &
797    &
798    &
799    &
800    &
801    &
802    &
803    &
804    &
805    &
806    &
807    &
808    &
809    &
810    &
811    &
812    &
813    &
814    &
815    &
816    &
817    &
818    &
819    &
820    &
821    &
822    &
823    &
824    &
825    &
826    &
827    &
828    &
829    &
830    &
831    &
832    &
833    &
834    &
835    &
836    &
837    &
838    &
839    &
840    &
841    &
842    &
843    &
844    &
845    &
846    &
847    &
848    &
849    &
850    &
851    &
852    &
853    &
854    &
855    &
856    &
857    &
858    &
859    &
860    &
861    &
862    &
863    &
864    &
865    &
866    &
867    &
868    &
869    &
870    &
871    &
872    &
873    &
874    &
875    &
876    &
877    &
878    &
879    &
880    &
881    &
882    &
883    &
884    &
885    &
886    &
887    &
888    &
889    &
890    &
891    &
892    &
893    &
894    &
895    &
896    &
897    &
898    &
899    &
900    &
901    &
902    &
903    &
904    &
905    &
906    &
907    &
908    &
909    &
910    &
911    &
912    &
913    &
914    &
915    &
916    &
917    &
918    &
919    &
920    &
921    &
922    &
923    &
924    &
925    &
926    &
927    &
928    &
929    &
930    &
931    &
932    &
933    &
934    &
935    &
936    &
937    &
938    &
939    &
940    &
941    &
942    &
943    &
944    &
945    &
946    &
947    &
948    &
949    &
950    &
951    &
952    &
953    &
954    &
955    &
956    &
957    &
958    &
959    &
960    &
961    &
962    &
963    &
964    &
965    &
966    &
967    &
968    &
969    &
970    &
971    &
972    &
973    &
974    &
975    &
976    &
977    &
978    &
979    &
980    &
981    &
982    &
983    &
984    &
985    &
986    &
987    &
988    &
989    &
990    &
991    &
992    &
993    &
994    &
995    &
996    &
997    &
998    &
999    &
1000   &

```


Run CP/M On Your DEC With The Bridge. Run inexpensive CP/M® programs on your DEC® minicomputer with The Bridge™ virtual microcomputer system. Now, every terminal attached to your minicomputer can run CP/M programs directly, as if they were dedicated microcomputers. Finally — the gap between the worlds of mini and micro disappears.

The Virtual Microcomputer System. The Bridge is a system of hardware and software for your DEC LSI-11®, PDP-11®, or VAX® minicomputer. The hardware, called the z-Board™, executes the program at speeds greater than those of dedicated micros. Together they constitute a breakthrough — unlocking a vast library of cost-efficient software for your minicomputer.

The z-Board. The z-Board is a hardware accelerator for The Bridge featuring four z-80's®,

256K bytes of memory, and a bit slice state machine. A key component of the system, the z-Board provides four users (per board) with an enhanced microcomputer environment. The z-Board is available for the Q-bus® and UNIBUS®.

System Integration. One of the key benefits of The Bridge is that all microcomputer operations are centralized in your mini. It's no longer necessary to maintain various micros at different work stations — all with different disk formats. The Bridge gives you one format, available to each operator, at every terminal. Every virtual micro can share files with every other virtual micro in your system.

Bridge Your Gap. Make the most of your DEC mini with The Bridge virtual microcomputer system. For more information, call Deryk Van Brunt today at (415) 841-9594.

virtual
microsystems

2150 Shattuck, Berkeley, CA 94704

CIRCLE 157 ON READER CARD

Bridge the mini/micro gap.



z-80 is a registered trademark of Zilog Corporation.
CP/M is a registered trademark of Digital Research.
The Bridge and z-Board are trademarks of Virtual
Microsystems, Inc.
DEC, LSI-11, PDP-11, VAX, Q-BUS, and UNIBUS are registered trademarks of Digital Equipment Corporation.


```

1000 I &
1001 I & GENERAL DESCRIPTION &
1002 I &
1003 FORMS is designed to allow the person responsible for maintaining &
1004 the proper forms in the line printers to change the type and the &
1005 parameters in use by a spooler without having to remember all the &
1006 specifics of form width and length, etc. &
1007 I &
1008 All of the forms used by an installation should be loaded &
1009 into the data file %FORMS.DAT using the format: &
1010 I &
1011 <formname>[/<switch1>[/<switch2>[etc....]]] I <description> &
1012 I &
1013 Ex: CHECK /DFLENGTH:22/LENGTH:22/WIDTH:80/HEADINGS:0 I Pay Checks &
1014 I &
1015 Note: All switches that could change should be specified, and the &
1016 exclamation point must be included. &
1017 I &
1018 The user then has only to type the number or name of the needed &
1019 form at the appropriate time. Form "0" may be specified to request &
1020 of the spooler the form it is currently using. &
1021 I &
1022 FORMS may be entered by "RUN %FORM" or by CCL entry. &
1023 The format for the CCL command is: &
1024 I &
1025 FORMS /<formname or number>[/<receiver ID or number> &
1026 I &
1027 The receiver ID need not be specified if only one spooler is in &
1028 operation. &
1029 I &
1030 The switch '/A' may be used with the form selection to &
1031 check the alignment of the paper. It causes the /ALIGN switch to &
1032 be sent to the spooler, and waits to send 'GO' or 'RETRY' back to &
1033 the spooler after the alignment pass is printed. &
1034 I &
1035 This program and data file may be moved to any desired &
1036 account without affecting program operation with the limitation &
1037 that the program and the data file reside in the same account. &
1038 I &
1039 If non-privileged operators are to use this program, it &
1040 must be protected <232>, and the CCL specification should include &
1041 the PRIV option. &
1042 I &
900 I &
901 I & DIMENSION STATEMENT &
902 I &
903 DIM M$(30), FORM$(10), LOGNAM$(10) &
904 I &
905 I Change FORM$( ) dimension to reflect the maximum number of forms &
906 I Change LOGNAM$( ) dimension to reflect the maximum number of spoolers &
907 I &
1000 I &
1001 I & START &
1002 I &
1003 ON ERROR GOTO 19000 &
1004 V$ = SYS(CHR$(65)+CHR$(75)) &
1005 DROP.PRIV$ = SYS(CHR$(65)+CHR$(215)+CHR$(15)) &
1006 ES$ = CHR$(155) &
1007 CLEAR.SCREEN$ = ES$ + '[721' + ES$ + 'H' + ES$ + 'J' &
1008 UP.CUR$ = ES$ + 'A' &
1009 CLEAR.EOL$ = ES$ + 'K' &
1010 CR.LF$ = CHR$(13) + CHR$(10) &
1011 SPACING$ = ' ' &
1012 SYSTEM$ = CVT$(RIGHT(SYS(CHR$(65)+CHR$(95)+CHR$(05)),3),4) &
1013 VER.ED$ = "V1.3" &
1014 NOTE: VT52 type screen commands forced (in CLEAR.SCREEN$) and used &
1015 I &
1030 CHANGE SYS(CHR$(125)) TO M$ &
1031 HOME$ = '_' + CHR$(M$(23)) + CHR$(M$(24)) + NUM$(M$(25)) + ':' &
1032 I &
1033 IF M$(26) AND 1$ &
1034 HOME$ = HOME$ + '[' + NUM$(M$(65)) + ',' + NUM$(M$(55)) + ']' &
1035 IF M$(35)+SWAP$(M$(50))<15%2$ &
1036 THEN &
1037 PRINT 'Please "RUN FORMS"' &
1038 GOTO 32767 &
1039 I &
1040 Build Device name, Account number of last opened file &
1041 If it was opened on channel 15 then it was us loading &
1042 Else, ask user to run us and exit &
1043 I &
1040 SNDLST$ = SWAP$(CVT$(MID(SYS(CHR$(65)+CHR$(125)),19,2),2)) &
1041 SPOOLERS$, OPSE$.THERE$ = 0$ &
1042 REGN.PRIV$ = SYS(CHR$(65)+CHR$(215)+CHR$(05)) &
1043 PTR$ = PEEK(SNDLST$) &
1044 IF PTR$ = 0$ &
1045 THEN &
1046 PRINT &
1047 PRINT 'No message receivers-OPSER not running' &
1048 GOTO 32767 &
1049 I &
1050 Get pointer to Receiver's list from 'Get Monitor Tables, II' &
1051 Initialize spoolers count, and OPSER running flag &
1052 Get pointer to first receiver &
1053 Quit now if there aren't any receivers &
1054 I &
1060 LOGNAM$ = ' ' &
1061 LOGNAM$ = LOGNAM$ + CVT$(SWAP$(PEEK(PTR$+OFFSET))) &
1062 I &
1063 FOR OFFSET$ = 2$ TO 6$ STEP 2$ &
1064 PTR$ = PEEK(PTR$) &
1065 OPSE$.THERE$ = 1$ IF LOGNAM$ = 'OPSE' &
1066 IF RIGHT(LOGNAM$,5) = 'SPL' AND LEFT(LOGNAM$,2) = 'LP' &
1067 THEN &
1068 SPOOLERS$ = SPOOLERS$ + 1$ &
1069 LOGNAM$(SPOOLERS$) = LOGNAM$ &
1070 I &
1071 Build the Receiver ID for this receiver &
1072 Get next receiver location &
1073 Flag OPSER as there if we encounter it &
1074 Otherwise if LOGNAM = 'LP7SPL' then increment counter and save &
1075 GOTO 1060 IF PTR$ &
1076 DROP.PRIV$ = SYS(CHR$(65)+CHR$(215)+CHR$(15)) &
1077 IF NOT OPSE$.THERE$ &
1078 THEN &
1079 PRINT &
1080 PRINT 'OPSE not running, can't process request' &
1081 GOTO 32767 &
1082 I &
1100 IF SPOOLERS$ = 0$ &
1101 THEN &
1102 PRINT &
1103 PRINT 'No print spoolers on line' &
1104 GOTO 32767 &
1105 I &
1120 IF SPOOLERS$ = 1$ &
1121 THEN &
1122 1$ = SPOOLERS$ &
1123 GOTO 1260 &
1124 I &
1125 If there's only one spooler, there's no choice &
1126 I &
1140 GOTO 1200 IF CCL$ AND LEN(SPOOLERS$) &
1141 PRINT CLEAR.SCREEN$ &
1142 PRINT 'FORMS';VER.ED$;TAB(13);SYSTEM$; &
1143 PRINT ' Spooler Menu' &
1144 I &
1145 Jump down to parse on CCL entry &
1146 Else clear screen and print header &
1147 I &
1180 PRINT %No such Spooler - ' ; SPOOLERS; CLEAR.EOL$ &
1181 PRINT UP.CUR$; UP.CUR$; CLEAR.EOL$; &
1182 SPOOLERS$ = ' ' &
1183 GOTO 1140 IF CCL$ &
1184 GOTO 1180 &
1185 I &
1186 Complain about errors in choosing spooler &
1187 I &
1260 SPOOLERS = LOGNAM$(1$) &
1261 Remember which spooler for the send to OPSE &
1262 I &
1280 PREFIX$ = CHR$(128+64) + 'INT' &
1281 OPSE command prefix to requests &
1282 I &
1400 I &
1401 R E A D I N F O R M S &
1402 I &
1420 REGN.PRIV$ = SYS(CHR$(65)+CHR$(215)+CHR$(05)) &
1421 OPEN HOME$ = '%FORMS.DAT' FOR INPUT AS FILE 1$, MODE 8192$ &
1422 LOSE.PRIV$ = SYS(CHR$(65)+CHR$(215)) &
1423 I &
1440 INPUT LINE #1$, V$ &
1441 FORM$ = FORM$ + 1$ &
1442 FORM$(FORM$) = CVT$(V$,132) &
1443 GOTO 1440 &
1444 I &
1445 Come here on FORMS.DAT end of file &
1446 I &
1460 CLOSE 1$ &
1461 I &
1700 I &
1701 F O R M S E L E C T I O N &
1702 I &
1720 GOTO 1760 IF CCL$ AND LEN(FORM$) &
1721 PRINT CLEAR.SCREEN$ &
1722 PRINT 'FORMS';VER.ED$;TAB(13);SYSTEM$; &
1723 PRINT ' Forms Menu Spooler: ' ; SPOOLERS$ &
1724 PRINT &
1725 SPACING$ = CR.LF$ IF FORM$ < 11$ &
1726 PRINT FORM$;TAB(5);FORM$(FORM$); SPACING$ FOR FORM$ = 1$ TO FORM$ &
1727 PRINT UNLESS LEN(SPACING$) &
1728 I &
1729 Jump down to parse if CCL entry &
1730 Else, clear screen, print header &
1731 Set up a blank line between form descriptions if there's room. &
1732 Display form options (more than 21 will overflow the screen) &
1733 I &
1740 INPUT 'Form <1>': FORM$ &
1741 FORM$ = CVT$(FORM$,38) &
1742 1$ = INSTR(1$,FORM$,'/A') &
1743 IF 1$ &
1744 THEN &
1745 ALIGN$ = -1$ &
1746 FORM$ = LEFT(FORM$,1$-1$) &
1747 I &
1760 FORM$ = '1' UNLESS LEN(FORM$) &
1761 FORM$ = VAL(FORM$) &
1762 IF FORM$ <= FORM$ &
1763 THEN &
1764 1820 &
1765 ELSE &
1766 FORM$ = 'Value too big: ' + FORM$ &
1767 GOTO 1800 &
1768 I &
1780 1$ = LEN(FORM$) &
1781 GOTO 1820 IF LEFT(FORM$(FORM$),1$) = FORM$ FOR FORM$ = 1$ TO FORM$ &
1782 I &
1783 Try for a match &
1784 I &
1785 NOTE: We're only trying to match up to the length of the form &
1786 request, i.e. A request of 'N' will match the first form &
1787 starting with 'N'. &
1788 I &
1800 PRINT 'No such form - ' ; FORM$; CLEAR.EOL$ &
1801 PRINT UP.CUR$; UP.CUR$; CLEAR.EOL$; &
1802 IF NOT CCL$ &
1803 THEN &
1804 1740 &
1805 ELSE &
1806 CCL$ = 0$ &
1807 GOTO 1720 &
1808 I &
1809 Complain about illegal forms and try again &
1810 I &
1820 PRINT CLEAR.EOL$ &
1821 1$ = INSTR(1$, FORM$(FORM$), '1') &
1822 1$ = LEN(FORM$(FORM$)) + 1$ UNLESS 1$ &
1823 REQUEST$ = 'FOR ' + LEFT( FORM$(FORM$), 1$-2$ ) &
1824 REQUEST$ = REQUEST$ + '/ALIGN' &
1825 IF ALIGN$ AND (INSTR(1$, FORM$(FORM$), '/ALIGN') = 0$) &
1826 ALIGN$ = (INSTR(1$, REQUEST$, '/ALIGN') <> 0$) &
1827 GOSUB 10000 &
1828 IF ALIGN$ &
1829 THEN &
1830 1840 &
1831 ELSE &
1832 V$ = SYS(CHR$(5)) &
1833 GOTO 32767 &
1834 I &
1835 Drop down a line, and clear any complaints while we're at it &
1836 Set up fully the request line
```

```

09998 RESUME 19999
09999 PRINT '7Unexpected FORMS error - ' ; &
PRINT CVT$(RIGHT(SYS(CHR$(6$)+CHR$(9$)+CHR$(ERR)),4$),4$) ; &
PRINT ' ' at line'; ERL &
PRINT ' ' at line'; ERL &
30000 I &
I & CCL ENTRY &
I &
30020 V$ = CVT$(SYS(CHR$(7$)), 8$+16$+32$+128$) &
IF LEFT(V$,5$) = 'FORMS' &
THEN &
ON ERROR GOTO 19000 &
V$ = RIGHT(V$,7$) &
GOTO 1020 UNLESS LEN(V$) &
CCL$ = -1$ &
ALIGN$ = INSTR(2$,V$, '/'A') &
V$ = LEFT(V$,ALIGN$-1$) + RIGHT(V$,ALIGN$+2$) IF ALIGN
1$ = INSTR(1$,V$, '/'') &
SPOOLER$ = RIGHT(V$,1$+1$) IF 1$ &
V$ = LEFT(V$,1$-1$) IF 1$ &
FORM$ = V$ &
GOTO 1020 &
I Start looking for '/'A' switch after 1st position &
because form name might start with 'A' &
30040 PRINT '?Illegal CCL entry' &
32760 CLOSE 1$
32767 END

```

```
NORMAL/DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:1 14x11(6) gbar, w/header
WIDE /DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:0 14x11(6), no header
NARROW/DFLENGTH:66/LENGTH:66/WIDTH:680/HEADINGS:0 8x11(6), no header
HEADED/DFLENGTH:66/LENGTH:66/WIDTH:100/HEADINGS:1 8x11(6), w/header
CHECK /DFLENGTH:22/LENGTH:22/WIDTH:180/HEADINGS:0/ALIG (6) Pay Checks
```

Ready

```

RUN %FORMS
<Clear Screen>
FORMS V1.2      RSTS V7.0-07 Oconee County Spooler Menu

1  LP0SPL
2  LP1SPL

```

Change forms on which Spooler <1>? <Default taken>

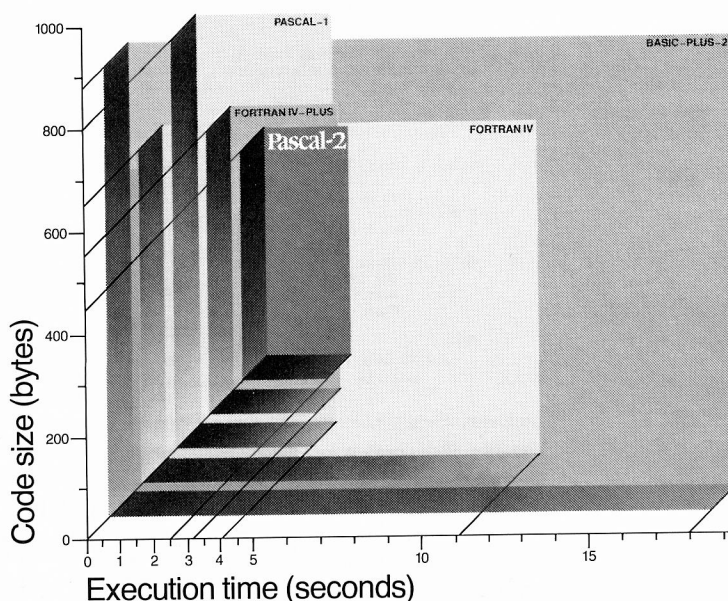
```
<Clear Screen>
FORMS V1.2 RSTS V7.0-07 Oconee County Forms Menu Spooler: LPOSPL

1 NORMAL/DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:1 14x11(6) gbar, w/header

2 WIDE/DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:0 14x11(6), no header
```

Pascal-2TM

PDP-11 RSX, RSTS/E, RT-11 and TSX-Plus



**Quicksort benchmark
executed on a PDP-11/45.**

The Pascal-2 system consists of an optimizing compiler, a high-level debugger and other utilities.

Write us for benchmark details.

Oregon Software

The Pioneer in Performance Pascal

2340 S.W. Canyon Road
Portland, Oregon 97201
(503) 226-7760
TWX: 910-464-4779

FORTRAN IV-PLUS, BASIC-PLUS-2, PDP, VAX, RSX, RSTS/E and RT-11 are trademarks of Digital Equipment Corporation. TSX-Plus is a trademark of S & H Computer Systems.

**THIS ISSUE OF THE
RSTS PROFESSIONAL
MARKS OUR THIRD
ANNIVERSARY.**

32767 END

PDP-11/ RSTS USERS

Our Generalized Reporting System (GRS)
is absolutely the most powerful information
retrieval and formatting tool available for
RSTS systems. It enables you to:

- Give your users ad-hoc access to your database with an English-like query facility
 - Extend the life of old applications instead of days
 - Develop new reports in minutes, instead of days
 - Free up your programmers for more important assignments
- GRS is easily interfaced to any RSTS file system or DBMS. And it's available on a one-month trial basis for only \$100!
- Can you really afford to be without GRS?
- Call or write today for free literature and licensing information.
- ENTERPRISE
TECHNOLOGY

etc

ENTERPRISE
TECHNOLOGY
CORPORATION

305 Madison Avenue
New York, NY 10165
(212) 972-1860
Telex 177959

CIRCLE 4 ON READER CARD

TERMINALS FROM TRANSNET

PURCHASE PLAN • 12-24 MONTH FULL OWNERSHIP PLAN • 36 MONTH LEASE PLAN

	DESCRIPTION	PURCHASE PRICE	PER MONTH		
			12 MOS.	24 MOS.	36 MOS.
* DEC	LA34 DECwriter IV Forms Ctrl.	\$1,095	\$105	\$ 58	\$ 40
	LA100 Letter Printer RO	1,995	190	106	72
	LA120 DECwriter III KSR	2,295	220	122	83
	LA120 DECwriter III RO	2,095	200	112	75
	LA12A Portable DECwriter	2,950	280	155	106
	VT100 CRT DECscope	1,695	162	90	61
	VT101 CRT DECscope	1,195	115	67	43
	VT125 CRT Graphics	3,295	315	185	119
	VT131 CRT DECscope	1,745	167	93	63
	VT132 CRT DECscope	1,995	190	106	72
	VT18XAC Personal Computer Option	2,395	230	128	86
TEXAS INSTRUMENTS	T1745 Portable Terminal	1,595	153	85	58
	T1765 Bubble Memory Terminal	2,595	249	138	93
	T1840 CRT	1,795	173	96	65
	T1785 Portable KSR, 120 CPS	1,795	173	96	65
	T1787 Portable KSR, 120 CPS	2,195	211	117	80
	T1810 RO Printer	1,695	162	90	61
	T1820 KSR Printer	2,195	211	117	80
		ADM3A CRT Terminal	595	57	34
LEAR SIEGLER	ADMS CRT Terminal	645	62	36	24
	ADM32 CRT Terminal	1,165	112	65	42
C-ITOH	CIT-101 CRT	1,525	147	82	55
	CIT-161 Color CRT	2,675	257	143	97
	CIT-427 Color Graphic CRT	3,095	297	165	112
TELEVIDEO	910 CRT Terminal	650	62	36	24
	925 CRT Terminal	850	82	46	31
	950 CRT Terminal	1,075	103	57	39
NEC SPINWRITER	Letter Quality, 7715 RO	2,695	259	144	98
	Letter Quality, 7725 KSR	3,195	307	171	115
GENERAL ELECTRIC	2030 KSR Printer 30 CPS	1,195	115	67	43
	2120 KSR Printer 120 CPS	2,195	211	117	80
EPSON	MX-80 F/T Printer	745	71	42	27
	MX-100 Printer	895	86	48	32
TIMEPLEX	E0400 4 Channel Stat Mux	1,525	147	82	55
	E0800 8 Channel Stat Mux	2,050	197	110	74

*DEC is the trademark of Digital Equipment Corporation.

FULL OWNERSHIP AFTER 12 OR 24 MONTHS • 10% PURCHASE OPTION AFTER 36 MONTHS

MICROCOMPUTERS

APPLE • COMMODORE • HP87 • DEC

ACCESSORIES AND PERIPHERAL EQUIPMENT

ACOUSTIC COUPLERS • MODEMS • THERMAL PAPER • RIBBONS • INTERFACE MODULES • FLOPPY DISK UNITS



TRANSNET CORPORATION

1945 ROUTE 22 • UNION, N.J. 07083 • (201) 688-7800
TWX 710-985-5485 800-526-4965 OUTSIDE N.J.

CIRCLE 28 ON READER CARD

ATPK.TEC

By Steve Roy, P.O. Box 284, Bloomfield, CT 06002

Hi there ATPK and TECO lovers! This one should make you all happy.. it's a TECO macro that performs substitution on a command file and then feeds it to ATPK. It's very useful for running a standard command file many times, and changing a few parameters in each run. It will substitute up to 61 different strings (called TOKENS) and provides for user definition of the string that flags the TOKENS (initially '*'). Needless to say, it doesn't harm the original command file.

Important note: The macro contains several '|' characters which may print as vertical bars, but are not to be confused with uparrow... they are ASCII 124. The following strings represent actual CONTROL characters: ↑A, ↑E, ↑N, ↑R, and ↑T. The others found in the file (^U and ^C) are carets followed by the letter, for obvious reasons. The file contains a help text on how to use it and on valid switches. The macro inputs substitution strings from the keyboard after being revoked. To get the help text on your terminal use: MUNG ATPK,/HELP

Have fun!

```
^AToken substitution TECO macro - V1 - 9/82
^A
!SCAN PARM! J@:S\HELP\ "S ^A
The proper method of invoking this TECO macro is:
MUNG name, [logfile = ] commandfile [switches]
```

Where "name" is the name of the macro (i.e., ATPK for ATPK.TEC) This TECO macro will alter the specified command file by changing all TOKENS to character strings entered by the user. A TOKEN is any occurrence within the command file of a character preceded by an asterisk (*). The legal characters that may follow the asterisk are 1 through 9, A through Z, and (lowercase) a through z. The macro can be instructed to change the asterisk to any other character string by specifying the switch: /*string

RSTS PROFESSIONAL

Box 361 • Ft. Washington, PA 19034-0361 • (215)542-7008

- ☐ PAYMENT ENCLOSED for one year's subscription (6 issues).
US: \$35 / Canada & 1st class: \$50 US /
All other countries, air mail: \$60, payable in US dollars.

- ☐ BILL ME for one year's subscription (6 issues).
☐ US / ☐ Canada or 1st class / ☐ Foreign.

Please send BACK ISSUES circled: V. 1, #1 V. 2, #3 V. 3, #2 V. 4, #1 V. 4, #4
☐ \$10 per issue enclosed. V. 2, #1 V. 2, #4 V. 3, #3 V. 4, #2 V. 4, #5
☐ Bill me for \$12.50 per issue. V. 2, #2 V. 3, #1 V. 3, #4 V. 4, #3

- ☐ Send me a RSTS PRO
Tee Shirt — \$6.95 ☐ S ☐ M ☐ L ☐ XL (Adult Sizes Only)

Name _____

Address _____

_____ Suite _____

City/State/Zip _____

Country _____ Phone () _____

FREE CLASSIFIED AD WITH SUBSCRIPTION!!

Your first 12 words are absolutely FREE, only \$1.00 per word thereafter.

SPECIAL

All 14 Back Issues for \$95.00. Payment with order.

V4.6

All nulls and occurrences of CONTROL/L are removed from the command file.

Once the substitutions have been made, an output file is created and named ATPKnn.TMP where 'nn' is the job number. Then TECO chains to ATPK to execute the command file. The macro appends the ATPK switch '/DEL', which deletes the temporary command file. Any switches that this macro does not recognize are passed to ATPK. Legal switches to this macro are:

```
/*string      Changes the macro substitution flag character from
               asterisk (*) to 'string'.

/NOEXECUTE    Tells the macro to perform the substitution only,
               without chaining to ATPK for execution. This switch
               may be abbreviated to: /NOEX

/NODELETE     Tells the macro NOT to have ATPK delete the temporary
               file that contains the substituted commands. This
               switch may be abbreviated to: /NODEL
```

NOTE: When entering replacement strings for the tokens, the legal delimiters between strings are comma and <CR> (carriage return). To terminate input of token-replacement strings, enter the string: //<CR> To enter a token string of '///' just make sure that it is followed by a comma and another string. Null strings are valid and may be entered as two consecutive commas, or a comma followed by a <CR>. The replacement string interpreter accepts strings for token-replacement in the following order: 123456789ABCDEFGHIJKLMNPOQRSTUVWXYZabcdefghijklmnopqrstuvwxyz...so *1 is replaced with the first string, etc. To change the input string delimiter from comma to some other character, enter the first string as a null (i.e., start with a comma), followed by the character that is to be used, followed by the true first string. For example, to change the delimiter to ';', start the input with: ;xxxx where xxxx is the first string. Needless to say, the first string may not be null, and the delimiter may not be changed to '/'.

The macro prompts for the next replacement string whenever a <CR> is typed, so the strings may all be entered on one line, or each be entered on separate lines. Other than <CR>, CTRL/U, CTRL/R, and CTRL/C, all characters of ASCII value less than 32 are ignored.

The substitution occurs in reverse alphabetical (and numeric) order, so, to the extent that higher numbers and letters may contain lower numbered (and lettered) tokens, the substitution is recursive.

```
^A^C'
!LOCATE SWITCHES, IF ANY! J @:S\ \ "S -1U0 I ZUO'
!EXTRACT LOGFILE NAME! J @Q0,1:S\ \ "S -D O, .XL O, .D QO:-QL-1UO'
!DEFAULT INFILE EXTENSION! J @Q0,1:S\ \ "U QOJ @I\ .CMD\ QO+4UO'
!SAVE AND OPEN THE INFILE! J .,QOXI .,QOD @ER\ ^EQI\
!CREATE THE TEMPFILE NAME! J EJ+100\ @^UO\ATPK\ 1,3:XO 0,3D
@: ^UO\ .TMP\
!OPEN IT & SETUP DEFAULTS! @EW\ ^EQO\ -1UD -1UE @^UF\ @^US\DEL\
!HANDLE SWITCHES NEXT! J @:S\ \ "U @OIREADY! '
O, .D @I\ \ ZJ@I\ \
!DEL! J @:S\DEL\ "S @FR\ \
!NODEL! J @:S\NODEL\ "S -6D OUD <@:S\^N\ \ ; -D> '
!NOEX! J @:S\NOEX\ "S -5D OUE <@:S\^N\ \ ; -D> '
!/*! J @:S\^*\ "S -2D @^UF\ \ <@:S\^N\ \ ; -1, .:XF -D> '
ZJ -D QD ^N @I\DEL\ \ HXS
!READY! HK 1UT
^AEnter replacement strings starting at:
^A
@^UT123456789ABCDEFGHIJKLMNPOQRSTUVWXYZabcdefghijklmnopqrstuvwxyz\
!PROMPT! HK GT QT-1, QTT ^A ? ^A HK OUR OUC
!I/O! ^T U1
!DEL! Q1-127 "E QC "G QC-1UC ET&2 "N 8^T^A ^A8^T
!QR "E ^A\^A' QC, QC+1T'
1^QRUR QC, QC+1D | ET&2&QR "N ^A\^A OUR ' '
@OII/OI ' OUR
!C/U! Q1-21 "E OUC OUR HK 13^T10^T @OII/OI '
!C/R! Q1-18 "E 13^T10^T HT OUR @OII/OI '
!RET! Q1-13 "E @OIPARSE! '
!OTHERS! Q1-31 "G OUR ZJ @QI1\ \ QC+1UC' @OII/OI
!PARSE! QT-1 "E J @:S\ \ "S 1, 2XD J 2D | @^UD\ \ '
ZJ GD < J @:S\ ^EQD\ \ ; -D O, .XQ [Q O, .D ^T-62 ; >
HK QT-61 "G 61UT @OIREPLACE! '
!CHECK//! GQ Z-2 "N @OIPROMPT! J @:S\ \ \ "U @OIPROMPT! ' JQ HK QT-1UT
!REPLACE! 5<A> ^AREplacing...
^A < QT-1UT QT "E :JQ "N ^A?Push-down stack not aligned^A^C' O;
'
J GT QT-1, QTXC O, .D JQ
< @:S\ ^EQF^EGC\ \ ; @FR\ \ GQ > >
!FINISH! EC QE "E ^C | @I\ATPK \ :QL "G GL @I\ \ GO GS
HXQ HK @EG\ ^EQQ\ '
!DEBUG @EI\ \
<ESC><ESC>
```

UNITRONIX MEANS DEC

**PDP 11/23
SYSTEMS**
with 2 Mbyte
memory, 160 Mbyte
winchester, std. 45 ips
9-track tape backup, CRT,
printer. Will run
RSX11, RSTS,
TSX or UNIX.
\$34,000

TERMINALS

VT100 LA12
VT101 LA34
VT102 LA50
VT125 LA100
VT131 LA120

CPUs

PDP 11/23,
11/44 & VAX

INTERFACES MODEMS

COMPLETE DATA SYSTEMS

with standard
and
custom
software



SPECIAL LSI-11 MODULES, PERSONAL COMPUTERS

WHY YOU SHOULD CALL UNITRONIX FIRST

- Immediate Delivery from \$6 mm Warehouse Inventory
- Complete Equipment Service Capability
- Excellent Pricing/Discount Structures

In just over 6 years, Unitronix Corporation has emerged as a leading supplier of Digital Equipment Corporation (DEC) microcomputers, terminal products, printers, ac-

cessories and supplies on both a domestic and international level. Our new, ultra-modern 20,000 sq. ft. facility in Somerville, New Jersey, houses executive offices, in-house programming, customer service, sales and marketing, component testing and complete warehouse facilities.

UNITRONIX

CORPORATION
197 Meister Ave., Somerville, NJ 08876
TELEX: 833184

AUTHORIZED
digital[®]
TERMINALS DISTRIBUTOR

CALL FOR LATEST PRICES
(201) 231-9400

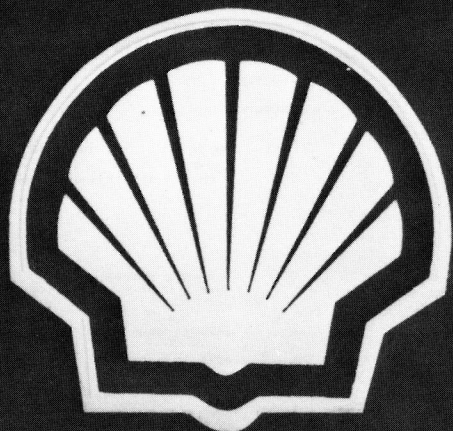
WRTOUT.BAS

By Raymond D. Tomlinson, Systems Engineer, Integrated Services, Inc., Aurora, CO 80012

```

11 SUB WRTOUT(DUM,DUM$,DUM%) TO BE USED FOR BASIC PLUS II,
12     DUM IS PASSED TO ROUTINE, DUM% IS
13     PASSED BACK TO BE PRINTED
14     AND DUM% IS A SWITCH FOR:
15     0% = THIS IS A DOLLAR/CENT PRINT
16     1% = THIS IS A DOLLAR PRINT ONLY
17     2% = THIS IS NOT A MONEY PRINT
18
21 DUMY = CTRLC      TRAP ^C WITHIN THIS ROUTINE
1  ON ERROR GOTO 32000 SET UP INTERNAL ERROR TRAPPING
3
*****
PROGRAM NAME:      WRTOUT.BAS
WRITTEN BY:        R. D. TOMLINSON
                   INTEGRATED SERVICES
                   1360 S. POTOMAC
                   AURORA, CO 80012
DATE WRITTEN:      JUNE 16, 1982
PURPOSE:           A SIMPLE CONVERSION
                   FOR CONVERTING A NUMBER
                   INTO A WRITTEN STRING.
*****
This is an APPENDable function to be used to
convert a real or floating point number into string
representation for printing on checks, receipts, etc.
*****
Until a library account can be established, this
routine will be residing in DR5:[1,37]WRTOUT.BAS.
*****
To use this routine, move it to your account
under the name you wish, delete lines 1 through 1000,
and APPEND it into your program. You may have to
change some line numbers, but I have tried to use some
which most programmers seldom use.
*****
To test this routine, OLD and RUN it. Enter
any positive real or floating point number between 0
and one billion. It will print the string that should
be on a check or receipt and ask for another number.
To terminate, enter a ^C.
*****
50 *****
C O P Y R I G H T
*****
Copyright (C) 1982 by
Integrated Services Inc., Aurora, Colorado
*****
This software is provided without charge by Integrated
Services, Inc. for use by any of the Integrated
Resources, Inc. companies and for distribution as they
should see fit.
*****
The information in this software is subject to change
without notice and should not be construed as a commit-
ment by Integrated Services, Inc.
*****
While all efforts have been made to remove problems
arising from the running of this program, there can be
no warranties nor guarantees, expressed or implied, by
ISI or the author resulting from the use of this
program on any system or computer.
*****
An attempt has been made to "round" the floating point
number within this routine; however, it is highly
advisable that before using, the number be rounded to
the programmer's, analyst's, or manager's desires. The
reasoning behind this advice is obvious to the
experienced programmer. You will see problems when
sending a number such as 123.4549999999999999 to the
routine, the PRINT USING will round to 46 cents while
the written version will round to 45 cents.
*****
60 *****
M O D I F I C A T I O N   H I S T O R Y
*****
VER/ED      EDIT DATE      REASON
V7.0-08C    07-SEP-82      Slight modifications of docu-
                             mentation lines to allow for
                             easier reading and understanding
                             and improved accuracy.
V7.0-08B    25-JUN-82      Add the function to allow no
                             dollars or cents to be printed
                             in the case where a whole number
                             is the item to be printed.
                             Also change the name to WRTOUT
                             from CHKWRT to broaden its
                             usefulness.
V7.0-08A    17-JUN-82      Allow up to .1E 13 -.01 to be
                             acceptable to the routine
                             and add printing of version #.
V7.0-08     16-JUN-82      Original Program written by
                             R. D. Tomlinson.
*****
75 OPEN 'KB:' AS FILE #1%
  MASK$(2%) = '###,###,###,###,###,###,###,###'
  \ MASK$(1%) = '##' + MASK$(2%)
  \ MASK$(0%) = MASK$(1%) + '.##'
*****
99 VERSION$ = 'V7.0-08C' + CHR$(9%) +
  RIGHT(SYS(CHR$(6%)+CHR$(9%)+CHR$(0%)),3%) +
  \ PRINT #1% VERSION$
100 INPUT #1% 'Amount 'A
  \ GOTO 100 IF A < 0.0
200 INPUT #1% 'Option 0,1,2,? 'A$
  \ GOTO 300 IF LEFT(A$,1%) = '?'
  \ A% = VAL(A$)
  \ GOTO 300 IF A% < 0% OR A% > 2%
  \ PRINT #1% USING MASK$(A%),A
  \ PRINT #1% FNWRTOUT$(A,A%)
  \ GOTO 100
300 PRINT #1% 'The Options are:'
  \ PRINT #1% '0 = Print "DOLLARS" and "CENTS"'
  \ PRINT #1% '1 = Print "DOLLARS" only'
  \ PRINT #1% '2 = Print the whole number, No "DOLLARS"'
  \ GOTO 200
29800 DEF FNWRTOUT$(DUM,DUM%)
  \ DUM$ = ''
  \ DUM = ABS(DUM)
  \ B,C,D = -1.
  \ DUM = DUM * 100 IF DUM% = 0%
  \ DUM = DUM + .5
  \ DUM = FIX(DUM)
  \ DUM = DUM / 100 IF DUM% = 0%
  \ THE "FUNCTION" WILL BE REMOVED FOR BP2
  \ SET SWITCHES FOR HUNDRED THOUSAND, MILLION, ETC.
  \ ROUND THE PASSED AMOUNT TO THE NEAREST PENNY
  \ OR WHOLE NUMBER DEPENDING UPON WHAT OPTION
  \ THE USER SELECTS.
29810 A1% = 1%
  \ A1% = 0% IF FIX(DUM) = 1.
  \ IF DUM < 1.0
  \ THEN
  \   DUM$ = DUM$ + 'ZERO '
  \   SET SWITCH (A1%) TO SHOW IF EXACTLY ONE DOLLAR
  \   SAY "ZERO" IF LESS THAN ONE DOLLAR
29820 GOTO 29920 IF DUM < 1.0 AND B < 0. AND C < 0. AND D < 0.
  \ IF DUM < 20.0
  \ IF DUM
  \ THEN
  \   DUM = DUM - 1%
  \   GOSUB 29970
  \   GOTO 29865 IF B > -1.
  \   GOTO 29875 IF C > -1.
  \   GOTO 29885 IF D > -1.
  \   GOTO 29920
  \ GO TO THE END IF WE ARE DONE
  \ OTHERWISE, IF THE AMOUNT LEFT IS LESS
  \ THAN 20 ADD THAT AMOUNT TO THE STRING
  \ AND REMOVE IT FROM THE BALANCE FORWARD
  \ (DUM) AMOUNT. IF OVER 100,000 THEN GO
  \ TO SOMEWHERE SPECIAL OTHERWISE GO TO
  \ THE END.
29830 IF DUM < 100.0
  \ THEN
  \   I% = DUM/10.0
  \   DUM = DUM - I% * 10%
  \   GOSUB 29980
  \   GOTO 29820
  \ IF THE BALANCE FORWARD (DUM) IS LESS
  \ THAN 100 THEN ADD THE "TENS" AMOUNT TO
  \ THE STRING, SUBTRACT FROM DUM AND START
  \ OVER.
29840 IF DUM < 1000.0
  \ THEN
  \   I% = DUM/100.0
  \   DUM = DUM - I% * 100%
  \   GOSUB 29970
  \   DUM$ = DUM$ + 'HUNDRED '
  \   GOTO 29820
  \ IF THE BALANCE FORWARD (DUM) IS LESS
  \ THAN 1000 THEN ADD THE "HUNDREDS"
  \ AMOUNT TO THE STRING, SUBTRACT FROM DUM
  \ AND START OVER.
29850 IF DUM < .2E 5
  \ THEN
  \   I% = DUM/1000.0
  \   DUM = DUM - I% * 1000.0
  \   GOSUB 29970
  \   DUM$ = DUM$ + 'THOUSAND '
  \   GOTO 29820
  \ IF THE BALANCE FORWARD (DUM) IS LESS
  \ THAN 20000 THEN ADD THE "TEENS" AMOUNT
  \ TO THE STRING, SUBTRACT FROM DUM AND
  \ START OVER.
29860 IF DUM < .1E 7
  \ AND B = -1.
  \ THEN
  \   B = DUM
  \   DUM = FIX(B/1000.)
  \   B = B - FIX(DUM * 1000.)
  \   GOTO 29820
  \ IF THE BALANCE FORWARD (DUM) IS LESS
  \ THAN 1,000,000 THEN ADD THE "THOUSANDS"
  \ AMOUNT TO THE STRING, SAVE EVERYTHING
  \ LESS AND START OVER.
29865 IF B > -1.
  \ THEN
  \   DUM = B
  \   DUM$ = DUM$ + 'THOUSAND '
  \   B = -1.
  \   GOTO 29820
  \ IF "B" WAS SET AT THE TEST FOR LESS
  \ THAN 1,000,000 THEN RESET DUM BACK TO
  \ "B" AND START OVER.
*****

```



**FPS80 HAS PROVEN ITSELF IN COMPANIES
LIKE SHELL INTERNATIONAL. IT IS BE-
COMING A BIG NAME IN FINANCIAL
PLANNING.**

JOIN US.

**WE ARE LOOKING FOR U.S. DISTRIBUTORS
FOR OUR PRODUCTS. NOW IS YOUR
CHANCE TO SHARE IN OUR SUCCESS.**

**IF YOU SELL DEC COMMERCIAL SOFT-
WARE, WHY NOT ADD THE UP AND COM-
ING FINANCIAL PLANNING PACKAGE TO
YOUR RANGE OF PRODUCTS?**

**FOR INFORMATION CONTACT:
D HOLROYD
RTZ COMPUTER SERVICES
6 ST JAMES'S SQUARE
LONDON SW1Y 4LD
TELEPHONE 01-930 4163
TELEX 24639**

DEAR RSTS MAN:

Paul M. Kvamme
Systems Programmer/Analyst
Northern Montana College

DEAR RSTS MAN:

Any file will be killed by LOGIN or LOGOUT whenever they are invoked if its

it until you close it (or chain, or end).
Thank you for an excellent magazine. It's
only fault is that it comes out too seldom.
Phil Rand
Automotive Wholesalers, Inc.
Kent, WA

```

THE NEXT INCREMENT OF EDITING. TO DO THIS THE
MAXIMUM AMOUNT OF A SHOULD BE INCREASED BY
A FACTOR OF 1000 AND A TRILLION BE THE TOP LEVEL

IX((DUM*100.)+.5)
DUM$(I%)
D' + I$ UNTIL LEN(I$) > 1%
DUM$ + I$+' '
THE DOLLARS ARE DONE, SO TACK THE CENTS ON TO
THE RETURNING STRING.

9990 IF DUM$
DUM$ + 'CENT'
S'
' IF I% = 1%
DUM$ + X$
ADD THE WORD "CENT(S)" TO THE STRING.

9990
THE ROUTINE IS NOW DONE SO GO BACK TO THE CALLER

DUM$ + 'ONE ' IF I% = 1%
DUM$ + 'TWO ' IF I% = 2%
DUM$ + 'THREE ' IF I% = 3%
DUM$ + 'FOUR ' IF I% = 4%
DUM$ + 'FIVE ' IF I% = 5%
DUM$ + 'SIX ' IF I% = 6%
DUM$ + 'SEVEN ' IF I% = 7%
DUM$ + 'EIGHT ' IF I% = 8%
DUM$ + 'NINE ' IF I% = 9%
DUM$ + 'TEN ' IF I% = 10%
DUM$ + 'ELEVEN ' IF I% = 11%
DUM$ + 'TWELVE ' IF I% = 12%
DUM$ + 'THIRTEEN ' IF I% = 13%
DUM$ + 'FOURTEEN ' IF I% = 14%
DUM$ + 'FIFTEEN ' IF I% = 15%
DUM$ + 'SIXTEEN ' IF I% = 16%
DUM$ + 'SEVENTEEN ' IF I% = 17%
DUM$ + 'EIGHTEEN ' IF I% = 18%
DUM$ + 'NINETEEN ' IF I% = 19%

THIS SUBROUTINE ADDS THE PROPER AMOUNT TO THE
RETURN STRING IF THE SUBJECT IS BETWEEN ONE
AND NINETEEN

DUM$ + 'TWENTY ' IF I% = 2%
DUM$ + 'THIRTY ' IF I% = 3%
DUM$ + 'FORTY ' IF I% = 4%
DUM$ + 'FIFTY ' IF I% = 5%
DUM$ + 'SIXTY ' IF I% = 6%
DUM$ + 'SEVENTY ' IF I% = 7%
DUM$ + 'EIGHTY ' IF I% = 8%
DUM$ + 'NINETY ' IF I% = 9%

THIS SUBROUTINE ADDS THE PROPER AMOUNT TO THE
RETURN STRING IF THE SUBJECT IS A "TEN"

DUM$ = DUM$
CALLER
IF THE FUNCTION - TO BE REMOVED FOR BP2

PROGRAM, CHANGE TO SUBEND FOR BP2, REMOVE FOR BAS

```

RESPOND

SOLVES YOUR RESPONSE TIME MYSTERY



TIME: Late

PLACE: Under the gun

CLUES: The RSTS/E* system's slow. Performance is under par. Users are unhappy. Hardware budget is gone. Conclusive evidence is lacking.

SOLUTION:

- RESPOND** is Manus' newest RSTS/E tuning tool.
- RESPOND** provides the information you need to tune your RSTS/E system.
- RESPOND** tells you where the bottlenecks are.
- RESPOND** measures response time exclusive of application program inefficiencies.
- RESPOND** logs statistics and jobs mix.
- RESPOND** reports tuning statistics in usable format.
- RESPOND** shows resource utilizations by program, keyboard and user account.
- RESPOND** ranks programs by resources used.
- RESPOND** reports long-term trends in summarized form.
- RESPOND** tells who did what, when and from which keyboard.
- RESPOND** warns you when free disk or small buffers are critical.
- RESPOND** logs out inactive terminals.

*RSTS/E is a registered trademark of Digital Equipment Corporation

YES!

*My system needs **RESPOND**
Send information kit.*

Name

Company

Address

Phone

MAKING COMPUTERS WORK FOR YOU

MANUS

SERVICES CORPORATION

AUTHORIZED
digital
COMPUTER DISTRIBUTOR

MANUS SERVICES CORPORATION

SEATTLE: 1700 Westlake Avenue North, 98109 • (206) 285-3260
PORTLAND: Kristin Square, Suite 102, 9498 SW Barbur Blvd. 97219 • (503) 245-1261

PROFIL

By Joe Blaho, Borden, Inc.
Chemical Information Systems
180 E. Broad St., Columbus, OH 43215

I. DESCRIPTION:

PROFIL.OBJ is a routine that, when linked into a task with BASIC+2 modules will count the number of times each line of the BASIC+2 program(s) is executed. When the program finishes or a !C is typed at the terminal a summary of each module/line and its execution count is displayed.

II. WHY USE IT:

This utility can be used when there is a need to optimize a program for performance. It will tell you which lines are executed most frequently. (The best solution would be to report how much time is spent at each line; this, however, would be much more complicated to do.) It might also be helpful if you have a program that is stuck in a loop and need to know where.

III. HOW TO USE IT:

To use this routine compile your program as usual; however, do not use the '/NOLINE' switch on any module you want a line count on. 'BUILD' the ODL and CMD files as follows:

```
HISEG      BP2COM,BASIC2 or NONE
ACCOUNT    LB:
BRL        NONE (ALWAYS !)
BUILD PGMA[,PGMB,PGMn],PROFIL
```

This will build your command files to link to the BP2COM library, (do not use a resident library IE: BASICS), and include the PROFIL routine along with your own routines. If you are using overlays make sure PROFIL is forced into the ROOT. Task build as usual. You should get some warnings about multi-defined symbols. This is expected and can be ignored. When the task builder finishes you may run your program. The results should be displayed when the program either ends, attempts to chain to another program, or you type !C.

IV. HOW IT WORKS:

A BASIC+2 program, when compiled with the /LINE option, generates a call to a routine named LIN\$ at the

start of each line. This routine saves the line number in a special area so that should there be an error, it would report the line number that was executing when the error occurred. PROFIL replaces this routine and intercepts this call from within the BASIC+2 program and counts how many times a unique module/line is executed. At the end of the program BASIC+2 calls a routine named END\$ (CHN\$ if a chain to another program is requested). These calls are also intercepted so we can display the results of the count.

V. NOTES:

a) Programs/sub-programs compiled with the '/NOLINE' option will not have their lines counted as the compiler does not generate calls to LIN\$.

b) The double integer display routine will display execution counts between 0 and 999,999,99. If you execute a line outside of this range the results are unknown (the program would probably not finish in our lifetime anyway).

c) If you exceed the maximum number of module names or lines the routine is assembled for (32 and 400) no new module names or line numbers will be counted; however, modules and lines already encountered will continue to be counted as they are executed.

d) Don't expect the program to run very fast when linked with this routine. PROFIL is intended to be used as an optimizing tool and the program should be re-linked without it when you are finished.

e) If the default table sizes (32 modules, 400 lines) are not acceptable or cause the routine to be too large, change the definitions of MAXLN or MAXNM to a more suitable number. For each line number counted it takes 8 bytes of storage, 2 bytes for each module name.

f) The 'CHAIN' statement is disabled and programs will not chain when this routine is used.

VI. EXAMPLES:

To Assemble PROFIL.OBJ:

```
MAC PROFIL,PROFIL = COMMON.MAC
, PRE.MAC , PROFIL.MAC
Get COMMON.MAC from RSTS distribu-
tion tapes, PRE.MAC from the BP2
tapes.
```

BASIC2

OLD TEST1

BASIC2

```
LIST
TEST1      03:14 PM      18-Oct-82
1          ! TEST1 - TEST OF PROFIL ROUTINE
200        FOR X=1. TO 100.
250                CALL TEST2
260                A=FND(X)
300        NEXT X
1000       DEF FND(S)
1005       PRINT S IF S/10. = INT(S/10.)
1010       FNEND
32767      END
```

BASIC2

COM/OBJ/LINE

BASIC2

OLD TEST2

BASIC2

```
LIST
TEST2      03:14 PM      18-Oct-82
1          SUB TEST2
300        FOR I= 1. TO 20.
325                X=X+1.
350        NEXT I
32767      SUBEND
```

BASIC2

COM/OBJ/LINE


BASIC2

```
HISEG
Name [NONE]--BP2COM
Account [LB:]--
BASIC2
BRL
File spec [LB:BASICS]--NONE
BASIC2
BUI TEST1,TEST2,PROFIL
BASIC2
TKB/SI:21 @TEST1
$TKB -- *DIAG*-MODULE $ERROR MULTIPLY DEFINES SYMBOL FLN$
$TKB -- *DIAG*-MODULE $ERROR MULTIPLY DEFINES SYMBOL LIN$
$Task exit status: ERROR
```

```
BASIC2
RUN TEST1
10
20
30
40
50
60
70
80
90
100
* PROFIL - V1.00 LINE USAGE SUMMARY *
```

```
LINE #      USAGE COUNT
*** MODULE NAME = TEST1 ***
1          1
200        101
250        100
260        100
1000       101
1005       100
1010       100
300        101
32767      1
*** MODULE NAME = TEST2 ***
1          100
300        2100
325        2000
350        2100
32767      100
```

BASIC2



DATATRIEVE and DECword/DP are registered trademarks of Digital Equipment Corp. WORD-11 is a registered trademark of Data Processing Design.

CIRCLE 57 ON READER CARD

```

1      .TITLE  BP2 - BP2 RUN TIME PROFILE
2      .IDENT  /1.03/
3      .SBTTL  RELEASE INFORMATION
4
5      ;
6      WRITTEN BY:  JOE BLAHO, BORDEN INC.
7                  CHEMICAL INFORMATION SERVICES
8                  180 E. BROAD ST. , COLUMBUS, OHIO 43215
9
10     ;
11     REVISIONS:
12
13     1.01  20-SEP-82      JMB ,  ADD DOUBLE INTEGER COUNTER CODE
14     1.02  01-OCT-82      JMB ,  ADD MULTI-MODULE CODE
15     1.03  21-OCT-82      JMB ,  "C TRAP
16
17     ;
18     DESCRIPTION:
19     THIS ROUTINE WHEN LINKED WITH A BP2 PROGRAM WILL REPLACE THE
20     LIN$ AND FLN$ ROUTINE IN THE OTS AND RECORD THE NUMBER OF TIMES
21     A PROGRAM PASSES THRU A GIVEN LINE NUMBER DURING EXECUTION.
22     THE END$ AND CHN$ ROUTINES ARE INTERCEPTED TO PRINT OUT THE
23     RESULTS OF THE RUN. THE RESULTS WILL ALSO BE DISPLAYED WHEN
24     A "C IS TYPED.
25     ** NOTE ** SINCE THE CHAIN (CHN$) ROUTINE IS INTERCEPTED ,
26     PROGRAMS WILL NOT CHAIN . THIS ROUTINE WILL OVERRIDE
27     "C TRAPS THE BP2 PROGRAM MAY ENABLE.
28
29     ;
30     .GLOBL  $OTS      ; OBJECT TIME SYSTEM
31     .GLOBL  $CHSTK     ; CHECKS FOR STACK OVR FLW
32     .GLOBL  $CLSAL     ; CLOSES ALL OPEN CHANNELS
33     .GLOBL  $NOREX     ; NORMAL EXIT TO DEFAULT RTS
34     .MCALL  $CCA$S     ; "C TRAP ENABLE MACRO (RSXMAC.SML)
35
36     ;
37     ;
38     ;
39     ;
40     ;
41     ;
42     ;
43     ;
44     ;
45     ;
46     ;
47     ;
48     ;
49     ;
50     ;
51     ;
52     ;
53     ;
54     ;
55     ;
56     ;
57     ;
58     ;
59     ;
60     ;
61     ;
62     ;
63     ;
64     ;
65
66     000620      MAXLN = 400.      ; MAXIMUM NUMBER OF LINES TO BE COUNTERED
67     001440      CNTOFF = MAXLN * 2.  ; OFFSET FROM LINE TABLE TO COUNTER TABLE
68     000040      MAXNM = 32.      ; MAXIMUM NUMBER OF MODULE NAMES
69
70     ;
71     ;
72     ;
73     ;
74     ;
75     ;
76     ;
77     ;
78     ;
79     ;
80     ;
81     ;
82     ;
83     ;
84     ;
85     ;
86     ;
87     ;
88     ;
89     ;
90     ;
91     ;
92     ;
93     ;
94     ;
95     ;
96     ;
97
98     ;
99     ;
100    ;
101    ;
102    ;
103    ;
104    ;
105    ;
106    ;
107    ;
108    ;
109    ;
110    ;
111    ;
112    ;
113    ;
114    ;
115    ;
116    ;
117    ;
118    ;
119
120     .SBTTL  LIN$ - COUNT LINES AS THEY ARE EXECUTED
121     .PSECT  PROFIL,RO,I,GBL,CON,REL
122
123     ;
124     ;
125     ;
126     ;
127     ;
128     ;
129     ;
130     ;
131     ;
132     ;
133     ;
134     ;
135     ;
136     ;
137     ;
138     ;
139     ;
140     ;
141     ;
142     ;
143     ;
144     ;
145     ;
146     ;
147     ;
148     ;
149     ;
150     ;
151     ;
152     ;
153     ;
154     ;
155     ;
156     ;
157     ;
158     ;
159     ;
160     ;
161     ;
162     ;
163     ;
164     ;
165     ;
166     ;
167     ;
168     ;
169     ;
170     ;
171     ;
172     ;
173     ;
174     ;
175     ;
176     ;
177     ;
178     ;
179     ;
180     ;
181     ;
182     ;
183     ;
184     ;
185     ;
186     ;
187     ;
188     ;
189     ;
190     ;
191     ;
192     ;
193     ;
194     ;
195     ;
196     ;
197     ;
198     ;
199     ;
200     ;
201     ;
202     ;
203     ;
204     ;
205     ;
206     ;
207     ;
208     ;
209     ;
210     ;
211     ;
212     ;
213     ;
214     ;
215     ;
216     ;
217     ;
218     ;
219     ;
220     ;
221     ;
222     ;
223     ;
224     ;
225     ;
226     ;
227     ;
228     ;
229     ;
230     ;
231     ;
232     ;
233     ;
234     ;
235     ;
236     ;
237     ;
238     ;
239     ;
240     ;
241     ;
242     ;
243     ;
244     ;
245     ;
246     ;
247     ;
248     ;
249     ;
250     ;
251     ;
252     ;
253     ;
254     ;
255     ;
256     ;
257     ;
258     ;
259     ;
260     ;
261     ;
262     ;
263     ;
264     ;
265     ;
266     ;
267     ;
268     ;
269     ;
270     ;
271     ;
272     ;
273     ;
274     ;
275     ;
276     ;
277     ;
278     ;
279     ;
280     ;
281     ;
282     ;
283     ;
284     ;
285     ;
286     ;
287     ;
288     ;
289     ;
290     ;
291     ;
292     ;
293     ;
294     ;
295     ;
296     ;
297     ;
298     ;
299     ;
300     ;
301     ;
302     ;
303     ;
304     ;
305     ;
306     ;
307     ;
308     ;
309     ;
310     ;
311     ;
312     ;
313     ;
314     ;
315     ;
316     ;
317     ;
318     ;
319     ;
320     ;
321     ;
322     ;
323     ;
324     ;
325     ;
326     ;
327     ;
328     ;
329     ;
330     ;
331     ;
332     ;
333     ;
334     ;
335     ;
336     ;
337     ;
338     ;
339     ;
340     ;
341     ;
342     ;
343     ;
344     ;
345     ;
346     ;
347     ;
348     ;
349     ;
350     ;
351     ;
352     ;
353     ;
354     ;
355     ;
356     ;
357     ;
358     ;
359     ;
360     ;
361     ;
362     ;
363     ;
364     ;
365     ;
366     ;
367     ;
368     ;
369     ;
370     ;
371     ;
372     ;
373     ;
374     ;
375     ;
376     ;
377     ;
378     ;
379     ;
380     ;
381     ;
382     ;
383     ;
384     ;
385     ;
386     ;
387     ;
388     ;
389     ;
390     ;
391     ;
392     ;
393     ;
394     ;
395     ;
396     ;
397     ;
398     ;
399     ;
400     ;
401     ;
402     ;
403     ;
404     ;
405     ;
406     ;
407     ;
408     ;
409     ;
410     ;
411     ;
412     ;
413     ;
414     ;
415     ;
416     ;
417     ;
418     ;
419     ;
420     ;
421     ;
422     ;
423     ;
424     ;
425     ;
426     ;
427     ;
428     ;
429     ;
430     ;
431     ;
432     ;
433     ;
434     ;
435     ;
436     ;
437     ;
438     ;
439     ;
440     ;
441     ;
442     ;
443     ;
444     ;
445     ;
446     ;
447     ;
448     ;
449     ;
450     ;
451     ;
452     ;
453     ;
454     ;
455     ;
456     ;
457     ;
458     ;
459     ;
460     ;
461     ;
462     ;
463     ;
464     ;
465     ;
466     ;
467     ;
468     ;
469     ;
470     ;
471     ;
472     ;
473     ;
474     ;
475     ;
476     ;
477     ;
478     ;
479     ;
480     ;
481     ;
482     ;
483     ;
484     ;
485     ;
486     ;
487     ;
488     ;
489     ;
490     ;
491     ;
492     ;
493     ;
494     ;
495     ;
496     ;
497     ;
498     ;
499     ;
500     ;
501     ;
502     ;
503     ;
504     ;
505     ;
506     ;
507     ;
508     ;
509     ;
510     ;
511     ;
512     ;
513     ;
514     ;
515     ;
516     ;
517     ;
518     ;
519     ;
520     ;
521     ;
522     ;
523     ;
524     ;
525     ;
526     ;
527     ;
528     ;
529     ;
530     ;
531     ;
532     ;
533     ;
534     ;
535     ;
536     ;
537     ;
538     ;
539     ;
540     ;
541     ;
542     ;
543     ;
544     ;
545     ;
546     ;
547     ;
548     ;
549     ;
550     ;
551     ;
552     ;
553     ;
554     ;
555     ;
556     ;
557     ;
558     ;
559     ;
560     ;
561     ;
562     ;
563     ;
564     ;
565     ;
566     ;
567     ;
568     ;
569     ;
570     ;
571     ;
572     ;
573     ;
574     ;
575     ;
576     ;
577     ;
578     ;
579     ;
580     ;
581     ;
582     ;
583     ;
584     ;
585     ;
586     ;
587     ;
588     ;
589     ;
590     ;
591     ;
592     ;
593     ;
594     ;
595     ;
596     ;
597     ;
598     ;
599     ;
600     ;
601     ;
602     ;
603     ;
604     ;
605     ;
606     ;
607     ;
608     ;
609     ;
610     ;
611     ;
612     ;
613     ;
614     ;
615     ;
616     ;
617     ;
618     ;
619     ;
620     ;
621     ;
622     ;
623     ;
624     ;
625     ;
626     ;
627     ;
628     ;
629     ;
6
```


LETTERS to the RSTS Pro

... continued from page 6

April 29, 1982

Greg Steinkuhler
TRT Telcommunications
P.O. Box 8876
Ft. Lauderdale, Fl. 33310

Dear Greg:

I saw your memo in THE DATA MANAGERS for Feb., 1982. I am glad that the TECO SIG is trying for a comeback. I think that you might be interested in reading my views about TECO.

I started in the DEC world on a PDP-8 about 15 years ago. The OS/8 line editor was good enough. About 1976 I started to get into RT-11 and made the unhappy discovery that the RT-11 'standard' editor was not the same as the OS/8 editor. In mid-1977 I started working on RSX-11M and once again I was faced with having to learn a different editor. Rather than learn that RSX editor I started using TECO. TECO seemed to be essentially the same editor whether I was using OS/8, RT-11, or RSX-11M. Also, TECO was available on RSTS/E and RSX-11D and TOPS-10. In other words I saw it as the true DEC Standard Editor. I didn't particularly like TECO, I wasn't turned on by its editing power, nor did I use much more than a fraction of its capabilities. I just didn't want to have to learn and remember 3 or 4 different editors.

As PCS grew from a one man operation to two, three, and four, I made TECO our one and only editor. VTEDIT came along in 1979 and our editing efficiency seemed to increase 100% over standard TECO.

Then came the rude awakening. When we got to 2 people TECOing on RSTS/E or RSX-11M things got slow. With 3 people, it was very slow. With 4 people it was impossible to get any work done. It seems that from V28 to V35 and V36 TECO itself had grown quite large. Using VTEDIT compounded the problem. Early in 1980 we dumped TECO and went to KED on RSX-11M and RSTS/E. As long as we stuck to the PDP-11 we had an editor that was operating system independent. We found that on RSX-11M and RSTS/E we could support 5 KED's with about the same throughput as 2 TECO's.

On our RT-11 systems I found that TECO v36 using VTEDIT is too big for the XM monitor and doesn't work right with the FB monitor if the USR swaps. Since we rarely use the SJ monitor, TECO had to be abandoned on RT-11 as well.

... to page 41

120	000134	005267	000774		30\$:	INC	NULM,N		; INCREMENT NUMBER OF LINES REFERENCED
121	000140	011322				MOV	(R3)	,(R2)+	; MODULE NAME INDEX TO FIRST WORD
122	000142	010122				MOV	R1	,(R2)+	; STORE THIS LINE NUMBER
123									
124	000144	005742			40\$:	TST	-(R2)		; BACK OFF POINTER
125	000146	062702	001440			ADD	#CINTOFF,R2		; R2 => COUNTER TABLE (LOW WORD)
126	000152	062712	000001			ADD	#1,	(R2)	; ADD TO LINE COUNT
127	000156	005542				ADC	-(R2)		; OVERFLOW INTO DBL INTGR (HIGH WORD)
128									
129	000160				50\$:	POP	R3		; RESTORE MESS
130	000162					POP	R2		
131	000164					POP	R1		
132	000166					POP	R0		
133	000170					CALL	CHSTK		; CHECK FOR STACK OVRFLW
134	000174	000134				JMP	@(R4)+		; EXIT
135									
137						.SBTTL	FINISH UP ROUTINE		
138	000176	104026			CCEND::	.TTIRST			; ^C ENTRY POINT
139	000200	104020				.TTTECH			; CANCEL ^O EFFECTS OF ^C
140									
141	000202				CHN\$::				
142	000202				END\$::	PUSH	R4		; SAVE R4
143	000204	012700	000670'			MOV	#PROMPT,R0		; R0 => PROMPT ADDRESS
144	000210	012701	000102			MOV	#PRMLN,R1		; R1 => PROMPT LENGTH
145	000214					CALL	PRINT		; PRINT THE BANNER
146	000220	012704	001140'			MOV	#NAMTBL ,R4		; R4 = TABLE OF MODULE NAME PTR'S
147	000224	016702	000704		55\$:	MOV	NULM,N,R2		; R2 = NBR OF LINES USED
148	000230	012703	001240'			MOV	#LNTBL ,R3		; R3=> LINE TABLE
149	000234	011400				MOV	(R4),R0		
150	000236	012037	001014'			MOV	(R0)+,@#MPNAM		; COPY MODULE NAME TO OUTPUT LINE
151	000242	012037	001016'			MOV	(R0)+,@#MPNAM+2.		
152	000246	012037	001020'			MOV	(R0)+,@#MPNAM+4.		
153	000252	012700	000772'			MOV	#MPRMPT ,R0		; R0 => MODULE NAME LINE BUFFER
154	000256	012701	000036			MOV	#MPRLN ,R1		; R1 = PRINT LENGTH
155	000262					CALL	PRINT		; PRINT CURRENT MODULE NAME
156									
157	000266	022314			60\$:	CMF	(R3)+,(R4)		; IS THIS LINE FOR CURRENT MODULE ?
158	000270	010103				BNE	65\$; NOPE
159	000272	012700	001030'			MOV	#OUTBUF ,R0		; GET READY TO SPACE FILL OUTPUT LINE
160	000276	012701	000014			MOV	#OUTLN/2.,R1		; LINE LENGTH / 2
161	000302	005741				TST	-(R1)		; DONT CLEAR THE <CR> <LF>
162	000304	012720	020040		62\$:	MOV	#8224.,(R0)+		; 2 SPACES
163	000310	077103				SOB	R1 ,62\$; LOOP UNTIL ALL SPACE FILLED
164	000312					PUSH	(R3)		; PUSH LINE NUMBER
165	000314					PUSH			; CLR HIGH WORD
166	000316	010601				MOV	SP ,R1		; R1 => ARG (HIGH WRD OF DBL INT)
167	000320	012700	001030'			MOV	#OUTBF1 ,R0		; R0 => BUFFER FOR STRING CVT
168	000324					CALL	CVTISD		; CONVERT DBL INTGR TO DECIMAL STRING
169	000330					POP			; RESTORE STACK POINTER
170	000332					POP			;
171	000334	010300				MOV	R3 ,R0		;
172	000336	062700	001440			ADD	#CINTOFF ,R0		; R0 => USE COUNT TABLE
173	000342					PUSH	(R0)		; PUSH LOW WORD ONTO STACK
174	000344					PUSH	-(R0)		; PUSH HIGH WORD
175	000346	010601				MOV	SP ,R1		; R1 => HIGH WORD OF DBL INT COUNT
176	000350	012700	001044'			MOV	#OUTBF2 ,R0		

SYMBOL TABLE

```
JFSYS      004000
JOBNUM=    000040
KEY         000400
KH.SIZ=    000006
K.ACH =    000002
K.ADP =    000001
K.CHG =    000002
K.DUP =    000003
K.LEN =    000001
K.NCG =    000000
K.NDP =    000000
K.NUM =    000000
K.OFF =    000004
LENGTH=    000030
LEN1 =     000102
LEN2 =     000106
```

Yours Truly,
Process Control Systems, Inc.
Michael E. Mazzoni, President

A cartoon illustration of a stack of coins. The stack is composed of several coins, with the top one clearly visible. The top coin has a sad face drawn on it, with large, downturned eyes and a small, frowning mouth. The stack is tilted slightly to the right.

DSKBLD

- For more information, attach your business card to this ad and send to:

European Distributor:
DCS Dialog Computer Systeme AG,
Maienstrasse 39, CH-8050 Zurich,
Switzerland

MANUS
SERVICES CORPORATION

MTMP5 = 000406	0.KSZ = 000034	PTTAB = 000130	SCARTA= 000076	USRPPN 000734
MTMP6 = 000410	0.LUN = 000006	PUADR = 000112	SCATPH= 000074	USRPRT 000736
MTMP7 = 000412	0.MAP = 000012	PULEN = 000114	SCAVAL= 000062	USRSP 000400
MTMP8 = 000414	0.MFY = 000001	PUBLT = 000116	SCRLUN= 000016	UOQFQ 000014
MTMP9 = 000416	0.MOD = 000000	P.BAD 177754	SDATA = 000030	UU. ACT 177761
NAMTBL 001140R 003	0.MODS= 000026	P.BPT 177756	SIZALA = 001576	UU. ASS 000012
NEDERR= 000073	0.NAM = 000010	P.CC 177770	SIZCHT= 000100	UU. ATR 177747
NEGLEN= 000246	0.NON = 000006	P.CRAS 177744	SIZERR= 000054	UU. ATT 000006
NKEYS = 000135	0.NRW = 000004	P.DEXT 177734	SIZHDR= 000154	UU. BCK 177765
NMPTR = 000002	0.NSP = 000001	P.EMT 177762	SIZIBF= 000400	UU. BYE 000005
NOBUPS= 000040	0.NUL = 000000	P.FIS 177742	SIZM = 000012	UU. CCL 177750
NODEVC= 000006	0.ORG = 000001	P.FLAG 177732	SIZMST= 000122	UU. CHE 000023
NONATH= 000175	0.OUT = 000001	P.FPP 177766	SIZOTS= 000046	UU. CHU 000010
NONPKY= 000226	0.REC = 000014	P.IOT 177760	SIZTBF= 000554	UU. CLN 000002
NOHACS= 000100	0.RED = 000003	P.ISIZ 177736	SLOT = 000134	UU. CNV 000024
NOHOCM= 000004	0.REL = 000003	P.HSIZ 177740	SLPST = 000400	UU. DAL 000014
NOSUCH= 000005	0.RMT = 000003	P.NEW 177750	SP = 000006	UU. DAT 177762
NOTAVL= 000010	0.SEQ = 000002	P.OFF 177732	SPSAP = 000340	UU. DEA 000013
NOTCLDS= 000007	0.SPA = 000000	P.RUN 177752	SQRERR= 000066	UU. DET 000007
NOTIMP= 000372	0.SPN = 000005	P.SIZE 177774	STATUS= 000146	UU. DIE 177760
NOTMTM= 000025	0.STR = 000004	P.STRT 177746	STCNT = 000156	UU. DIR 000017
NOTMTA= 000101	0.TMP = 000010	P.TRAP 177764	SUBARG= 000130	UU. DLU 000001
NOTOPN= 000011	0.TRM = 000000	P.2CC 177772	SUBCOD= 177772	UU. ERR 000011
NOTUSE= 000144	0.TRN = 000004	Q\$\$\$ = 000002	SUBERR= 000067	UU. FCB 177770
NSTORG 000100	0.TSK = 000020	RABLEN= 000120	SUBREG= 000373	UU. HCG 177767
NULHND 000026	0.UDF = 000000	RABSTR= 000004	SUBSAY= 000322	UU. JOB 000030
NUMLN 001134R 003	0.UNS = 000005	RYBTES= 000134	SYMTBL= 000166	UU. LFN 000004
NUMNAM 001136R 003	0.VAR = 000002	RECADR= 000032	SYSVEE 000103	UU. LOG 177755
ODD = 000071	0.VFC = 000003	RECAD2= 000036	SYSLVZ 030600	UU. LOK 000021
ONBAD = 000072	0.VIR = 000001	RECLCN= 000030	TEMP1 = 000124	UU. MNT 000003
ONERDB= 000346	0.WFI = 000002	RECOUM= 000030	THREAD= 000224	UU. NAM 177776
ONERGO= 000110	PKACK= 000026	REWFI 000010	TLCNTD= 000155	UU. NLG 177757
OPNFQF 000002	PBFPT= 000152	RESERR= 000150	TLNOMF= 000141	UU. PAS 000000
OPNFS = 000030	PC = 000000	RESF\$ = 000150	TLTMFA= 000131	UU. POK 177772
OPSYS = 000041	PF.CSZ 020000	RJEHND 000024	TMFBKS= 000070	UU. PRI 177763
OP.SIZ= 000032	PF.EMT 100000	RMBERG= 000200	TMPLBL= 000066	UU. RAD 000016
OUTBF1 001030R 003	PF.KEM 000400	RMSERR= 000377	TMPI 001130R 003	UU. RTS 177756
OUTBF2 001044R 003	PF.NBR 004000	RMSIF\$ = 000000	TMPHDR= 000440	UU. SLN 000025
OUTBUF 001030R 003	PF.REM 010000	RNDMH = 000174	TMPL 001132R 003	UU. SWP 000027
OUTF\$ = 000010	PF.RW 020000	RNDML = 000172	TMPLMBF= 000072	UU. TB1 177775
OUTLN = 000030	PF.1US 001000	RSTFQ 000020	TMPORG= 000065	UU. TB2 177764
0.ACC = 000002	PKKBND 000020	RSTSYS= 000004	TMPRFM= 000064	UU. TRM 000020
0.ALW = 000004	PLTHND 000034	RKSXSY= 000001	TMPI1= 000050	UU. YLG 177777
0.APP = 000005	POSITN= 000146	RDXHND 000022	TMPIW13= 000054	UU. ZER 000015
0.BKS = 000022	POSITX= 177776	RO = 000000	TOPIOB= 000016	VCAERR= 000053
0.BLK = 000024	PRERRS= 000164	R1 = 000001	TOPSTR= 000012	VCCERR= 000055
0.BUF = 000026	PRINT 000616R 003	R2 = 000002	TPSSYS= 000005	VCCERR= 000052
0.CLU = 000020	PRIVAT= 000030	R3 = 000003	TRCL = 000162	VDF = 000006
0.CON = 000030	PRMLN = 000102	R4 = 000000	TRCP = 000162	VHERIT= 020000
0.CTG = 000002	PROMPT 000616R 003			

XRBLK	000010	X2	001110R	003	\$\$\$MAXC=	000016	.MESAG	104060	.STAT	104040	
XRBLKM	000007	X3	001104R	003	.CCL	104062	.NAME	104044	.TIME	104030	
XRSISZ	000016	X4	001100R	003	.CHAIN	104070	.PEEK	104012	.TIME	104016	
XRCI	000006	X5	001074R	003	.CLEAR	104038	.FOSTN	104032	.TTDIT	104024	
XRLN	000000	X6	001070B	003	.CORE	104036	.READ	104000	.TTECH	104020	
XRLC	000004	X7	001064R	003	.DATE	104034	.RTS	104050	.TTMCH	104022	
XRMOC	000014	X8	001060R	003	.ERLOG	104052	.RUN	104042	.TTRST	104026	
XRTIME	000012	XCLSAL=	*****	G	.EXIT	104046	.SET	104036	.UUG	104066	
X0	001120R	003	\$.NOREX=	*****	G	.FSS	104064	.SLEEP	104010	.WRITE	104004
X1	001114R	003	\$.OTSYS=	*****	G	.LOGS	104054	.SPEC	104014		

```

. ABS. 177777 00
        000000 00
BP2OTS 000000 00
PROFIL 007440 00
ERRORS DETECTED: 0

```

VIRTUAL MEMORY USED: 5267 WORDS (21 PAGES)
DYNAMIC MEMORY: 6082 WORDS (23 PAGES)
ELAPSED TIME: 00:00:39
PROFIL,PROFIL=COMMON,PRE,PROFIL

*RSTS/E is a registered trademark of Digital Equipment Corporation.

CIRCLE 160 ON READER CARD



DILOG INTERFACES WINCHESTER/BACK-UP PDP-11*

You can't find a better, high-performance line of low-cost, single board, disc and tape controllers for your PDP-11 data storage system. All Winchester disc and 1/2" tape controllers are based on uP architecture and automated design common to the thousands of DILOG controllers in use. So they consistently offer you best price/performance.

The vital time-saving and data base protecting features you want, including diagnostic test/self-test/error correction/etc., are all built into each intelligent controller. You'll also find RT-11, RSX-11, RSTS, IAS, MUMPS, DSM or VMS software compatibility, plus I/O compatibility with all popular drives.

WINCHESTER/SMD/CMD CONTROLLERS

These controllers mate with larger capacity hard disc drives employing RK, RM and RP emulations. The following is a partial listing of models:

NEW—UNIVERSAL SMD INTERFACE COMPATIBLE—Model

DU215—Interfaces UNIBUS to SMD I/O drives—switch selectable RK06/RK07—two 8" or 14" Winchester SMD/CMD drives to 300 MB capacity (mix or match without any component change)—56-bit ECC—88% utilization of unformatted storage capacity—exclusive universal formatting.

NEW—SMD INTERFACE COMPATIBLE—Model DU218—Full

RM02/RM05/RP06 media compatibility—interfaces PDP-11/04 through 11/60 computers with up to four Winchester, or CDC9762/9766 equivalent disc drives employing

industry standard SMD interface, as well as Memorex 677/RP06 type drives—includes three sector buffering, 32-bit ECC and dual port capability.

MAGNETIC TAPE CONTROLLERS

These 1/2" mag tape controllers/couplers handle standard start-stop and streaming drives with TM-11 or TS-11 emulations.

1/2" STREAMING/START-STOP COMPATIBLE—Model DU130—

Handles up to eight 9-track, 800/1600 BPI industry standard drives at 12.5 ips to 125 ips—interfaces dual density (NRZI/PE) drives from all manufacturers. TM-11 emulation.

NEW—1/2" MAGNETIC TAPE COMPATIBLE—Model DU132—

Emulates TS-11—interfaces up to four start-stop or streamer drives, including "Cache Streamer"**-12.5 ips to 125 ips speeds—exclusive 800/1600 BPI dual density—also interfaces GCR transports.

Contact your local DILOG sales office for complete details and O.E.M. quantity discount pricing/delivery of specific models providing Winchester/Backup for PDP-11.

Corporate Headquarters

12800 Garden Grove Blvd. • Garden Grove, Calif. 92643
• Phone: (714) 534-8950 • Telex: 681 399 DILOG GGV

Eastern Regional Sales Office

64-A White Street • Red Bank, New Jersey 07701
• Phone: (201) 530-0044

European Sales/Service Office

12 Temple Square • Aylesbury, Buckinghamshire • England
• Phone: 44-296-34319 or 34310 • Telex: 837 038 DILOG I G

**DISTRIBUTED
LOGIC CORP.**
DILOG
NUMBER 1 FOR DEC-11

*DEC, PDP, RSX, RSTS, IAS, VMS, UNIBUS Trademarks Digital Equipment Corp.
**Trademark Cipher Data Products

ELECTRONIC MAIL. PRACTICALLY SPEAKING.

Sooner or later you will be using electronic mail. It just makes good sense. When you do, you will want a system that is complete—a delivery system, a scheduling system, and an information manager. Your electronic mail system will become an essential part of your office environment. USER-MAIL is such an electronic mail system*.

USER-MAIL's power is easy to control. It relates to the way you work. Electronic IN, OUT, and HOLD baskets are just what you would expect. You can scan your IN basket, selecting only those message subjects you wish to read. Or, you can place a message into your HOLD basket for a number of days to have it automatically reappear in your IN basket on the appointed day. You can even have USER-MAIL recall specific messages by providing your own selection criteria. Replying, forwarding, and sending to groups are as easy as can be. And these are just a few of the features in store for you.

You owe yourself a closer look.
Write for a brochure or give
us a call direct.



**North County
Computer Services, Inc.**
2235 Meyers Ave.,
Escondido, California 92025
(714) 745-6006, Telex: 182773

*USER-MAIL is currently available on DEC computers using the RSTS operating system.
RSTS is a registered trademark of Digital Equipment Corporation.
USER-MAIL is a trademark of Logic eXtension Resources.



You can get more from your VAX computer or RSTS system with MAS-M.

MAS-M is the application software system from Martin Marietta Data Systems that can help you do more with your DEC hardware. That's because MAS-M is the on-line software system that gives you much more than you'd expect from packaged software.

More Flexibility.

MAS-M's modular design lets you choose from 10 different application systems:

- Accounts Receivable
- Accounts Payable
- General Ledger
- Order Processing
- Invoicing
- Inventory Control
- Inventory Accounting
- Bill of Materials
- Material Requirements Planning
- Purchasing

You can implement just the modules you need to satisfy your demands. And no matter which combination you choose, the MAS-M system is always fully integrated.

MAS-M's flexible design also makes it easy to install, and simple for your users to operate. The MAS-M/VAX package is

written in native mode VAX-11 BASIC and the MAS-M/PDP package is written in BASIC-PLUS-2 under RSTS/E. Both of these packages are based on the RMS data management system. These features make both MAS-M packages fully compatible with your current RSTS/E or VAX/VMS operating system.

More Control.

You can count on MAS-M for more comprehensive data accuracy and security, too.

MAS-M's powerful transaction processing MONITOR gives you maximum control over your data—from start to finish. User passwords and menu selections are checked against user security profiles. Data entry validation is also standardized in the MAS-M MONITOR, so any invalid data can be corrected *before* it reaches your application program.

More Productivity.

MONITOR is also an important tool in developing new applications. You can use MONITOR to create input screens and validation rules on-

line. And, MONITOR can help you improve programmer productivity by providing a standard framework for input of code that minimizes the difficulties of user interface and terminal characteristics.

More Support.

You can count on Martin Marietta Data Systems for system development and implementation, comprehensive training, and clear, concise documentation. We can also provide an extensive Maintenance Service to support your MAS-M system.

What it all adds up to is a packaged software system that can give you everything you need to get your jobs done. And more. Write or phone us today, and we'll tell you more about how the MAS-M software system can work for you.

MAS-M

The Software System That Can Help You Do More.

**MARTIN MARIETTA
DATA SYSTEMS**

Martin Marietta Data Systems
Marketing Services, R/H
6303 Ivy Lane, Greenbelt, MD 20770
(800) 638-7080 In Maryland (800) 492-7170

MARTIN MARIETTA

The image shows an IBM 5150 PC system. The monitor displays a text-based interface with a table of data. The table has columns for 'Date', 'Time', 'Location', 'Status', 'Action', 'Priority', 'Severity', 'Impact', 'Risk', 'Cost', 'Benefit', 'Value', 'Score', 'Rank', 'Order', 'Index', 'ID', 'Key', 'Code', 'Type', 'Category', 'Group', 'Subgroup', 'Item', 'Detail', 'Description', 'Explanation', 'Reason', 'Cause', 'Effect', 'Result', 'Outcome', 'Consequence', 'Implication', 'Significance', 'Importance', 'Relevance', 'Usefulness', 'Value', 'Worth', 'Price', 'Cost', 'Expense', 'Investment', 'Expenditure', 'Outlay', 'Payment', 'Contribution', 'Revenue', 'Income', 'Profit', 'Gain', 'Loss', 'Expense', 'Cost', 'Price', 'Value', 'Worth', 'Price', 'Cost', 'Expense', 'Investment', 'Expenditure', 'Outlay', 'Payment', 'Contribution', 'Revenue', 'Income', 'Profit', 'Gain', 'Loss'.

Whether you buy, rent or lease our equipment, you'll find MTI is the one source for all the terminals, peripherals, systems, applications expertise and service you'll ever need. At good prices. Call us today.

The logo for MTI Systems Corp. features the lowercase letters "mti" in a large, bold, sans-serif font. Below "mti", the words "systems corp." are written in a smaller, lowercase, sans-serif font. The entire logo is contained within a black, rounded rectangular background.

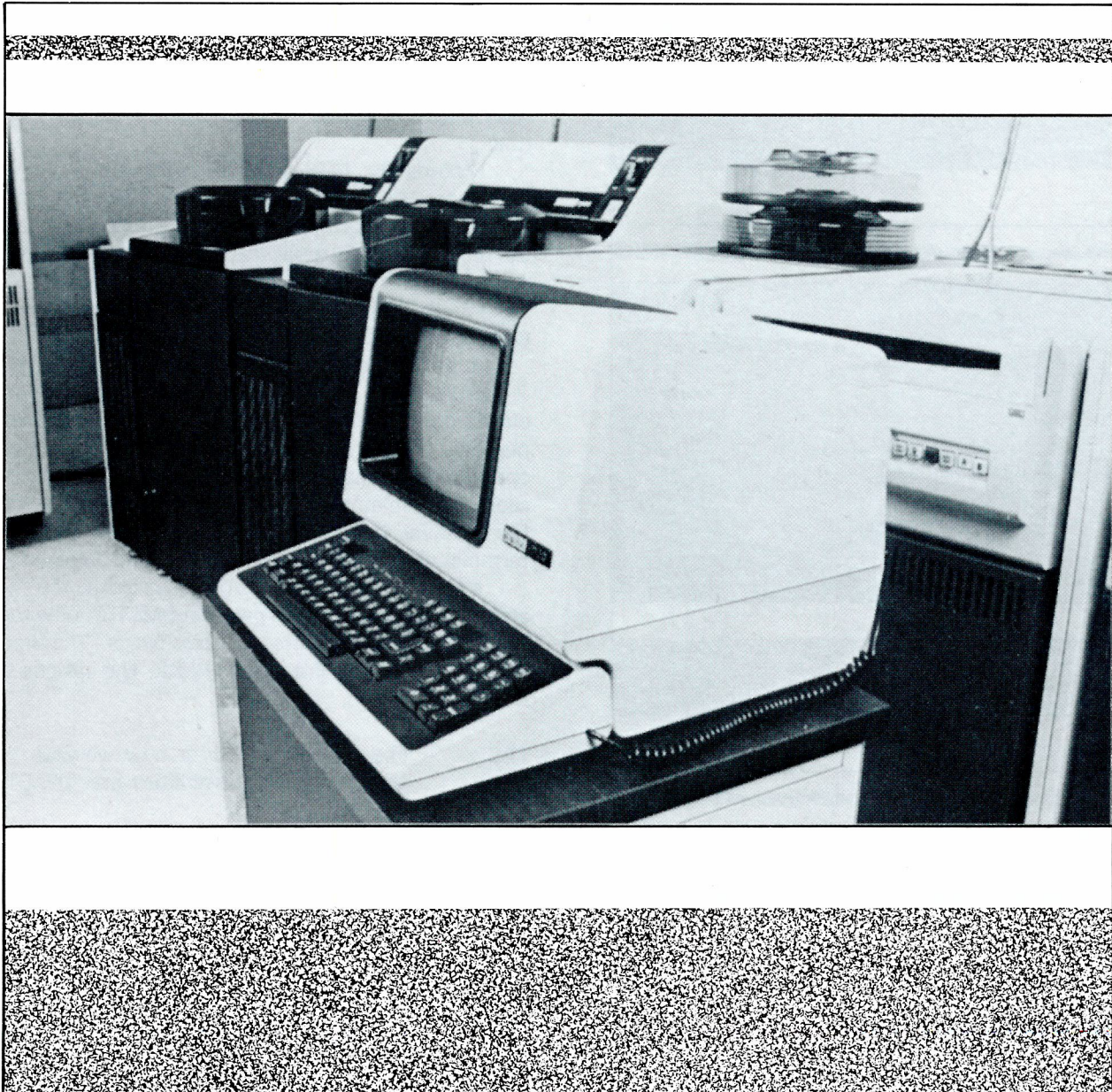
CIRCLE 107 ON READER CARD

The VAX-SCENE

Number 11

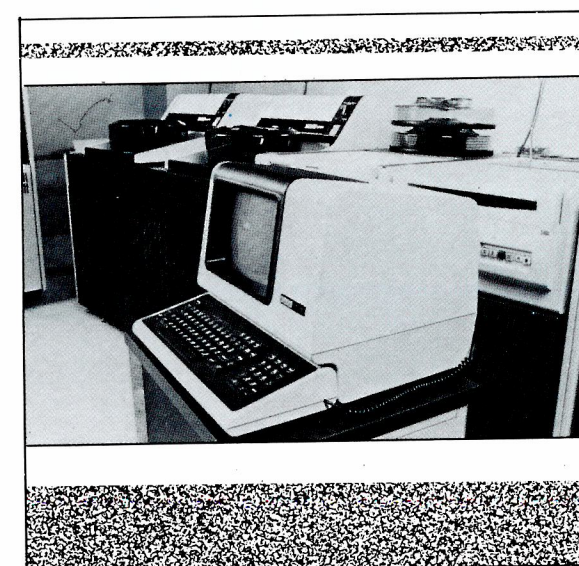
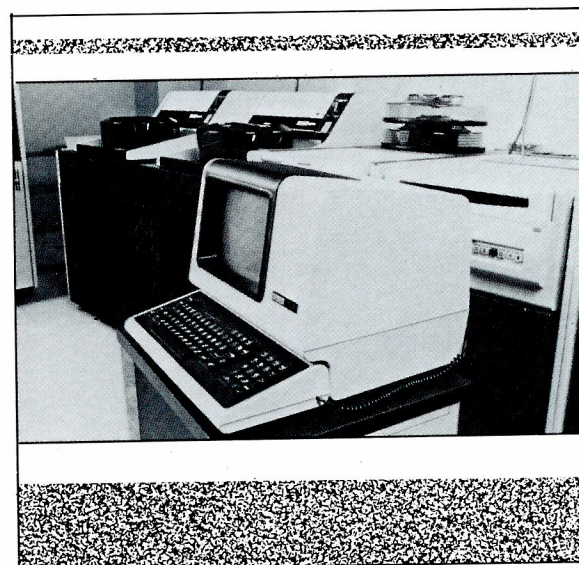
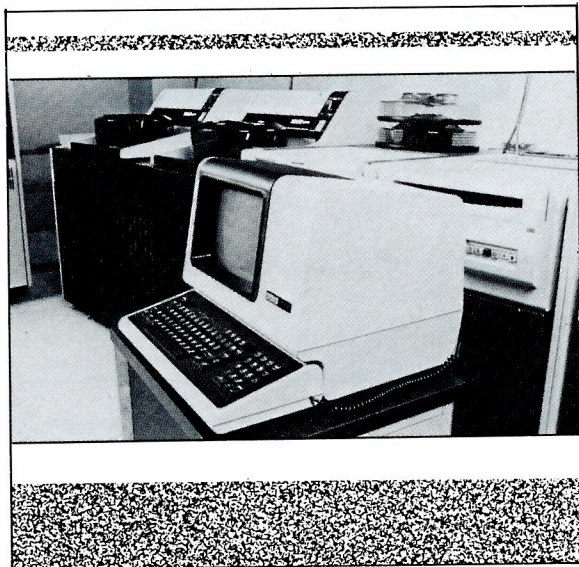
(RSTS PROFESSIONAL, Vol. 4, No. 6)

December 1982



INSIDE:

Reading RMS Files from VAX MACRO



Reading RMS Files from VAX MACRO

By Bob "MACRO MAN" Meyer

Long ago and far away, I attempted to use some RMS macro on a RSTS system. (I'm not sure exactly what possessed me to do this . . .). It was a simple program, in fact the program I tried to use was one that was given as an example in the back of the RMS-11 MACRO-11 Reference Manual. After several days and much bloodshed, I surrendered; the program simply would not work. I tried a few other such programs, but to no avail; I was convinced that RMS macro was just too complicated for poor ol' macro man . . .

Fortunately, RMS macro programming isn't quite so bad on the VAX. In fact, it actually works! The program included in this article is a simple example of how to define the required control areas (rabs & fabs), open an already existing file (this program); read records from that file (the statements in this program); and print them on the current terminal.

If you'll turn to the sample program, I'll narrate . . .

First, in the psect TTAREA, we define the name of our output device, TT:. Next we block off a word for the channel number; this number is assigned to us by VMS when we open the terminal (see the \$assign__s call in the CODE psect). This channel number is used in all write operations to the terminal. Now we define a macro to do printing to the terminal (PRINT). We'll be using this one a lot.

Moving to the psect FIBSTUFF, we define the two major control blocks required to perform RMS I/O. (The macro call \$FIBDEF defines all related symbols for us.) The first control block is the FAB or File Access Block. The options specified here are:

FAC	File access method; or how we wish to access the file. Some possibilities are: DELETE, GET, PUT, etc . . .
FNA	File name address; this is a pointer to the .ASCII file name.
FNS	File name size; this is the size of the string pointed to by FNA.
MRS	Maximum record size; this is the maximum size of any record in the file.
ORG	File Organization; a one byte field specifying the type of file to be accessed. This can be: SEQ,REL, or IDX.
RFM	Record format; some choices are: Fixed, Stream, Undefined, Variable, VFC.

Now we define the FIB or File ID block, and its descriptor (FIB_DESCR:) which points to the FIB block, and specifies its length.

Next we define the other major block, the RAB or Record Access Block. Some options used here are:

FAB	This is a pointer to the File Access Block. This field is used during \$CONNECT operations.
-----	---

RAC Record access field; in this case, sequential.

RBF	This field is a pointer to a record buffer.
-----	---

RSZ Size of the RBF field.

UBF Pointer to a user buffer (this is where the records will actually reside after a \$GET operation).

USZ Size of the UBF field.

Next we actually define the user buffer & record buffer. After these we have the full name of the file we wish to access, and the length of this filespec.

Finally we define some text & messages to make the output look presentable.

Now we start the actual code (at the label `READ:`). The first word in the program is a register save mask. Specific bit patterns in this word instruct the system to preserve certain registers upon entry into the routine. Since we're not interested in saving anything, we'll just set this to zero.

Next we assign (or open) the current terminal with the \$ASSIGN__S macro. The system returns a channel number to us (in TTCHAN) which we will use for all output (or input) to the terminal. The instruction BLBS says 'Branch if low bit set in R0 to 5\$'; since the low order bit is set after monitor calls to indicate success, this will cause a branch to the label 5\$ if the \$ASSIGN__S works properly. Otherwise, we just RET (return) to VMS with the error code in R0, and an error message will be printed on our terminal.

Assuming the \$ASSIGN__S works, we're ready to open the disk file. This is done with the \$OPEN macro. We simply specify the FAB in the open statement, and let 'er rip . . . If anything goes wrong, the BLBS instruction will, again, return control to the system, and an appropriate error message will be printed on the terminal.

At the label 10\$, we \$CONNECT the FAB with the RAB, making I/O to and from the file possible.

Next at 20\$, we start reading records. The \$GET macro gets the next (sequential) record from the file and places it in the user buffer specified in the RAB layout. After each \$GET we check for an error code in R0. If we have an end of file indication, we simply close the file and exit. If something else goes wrong, we blitz, since we're not really expecting any other errors.

If we indeed get a valid record, (40\$) we get the record length in R2, print the record (between quotes) on the terminal, and finally print a <CR> <LF> to make it readable. After all that we loop back to 20\$ and continue until we hit

end of file.

That's about all for now; have fun . . .

```

;title read
;ident /1.0/

;
;read.mar
;uses rms macro to read disk files
;

.psect ttarea
tt: .asciid /tt/
ttchan: .word 0

.macro print buf,siz
    $qiow_s chan=ttchan,-
    func=$io$_writevblk,-
    p1=buf,-
    p2=siz
.endm print

.psect fibstuff
;
;layout the fab & rab
;
;$fibdef
fab_block:
    $fab
        fac=get,-
        fna=file_name,-
        fns=file_size,-
        mrs=512,-
        org=seq,-
        rfm=fix

fib_block:
    .blkb fib$k_length

fib_descr:
    .long fib$k_length
    .long fib_block

rab_block:
    $rab
        fab=fab_block,-
        rac=seq,-
        rbf=record_buffer,-
        rsz=512,-
        ubf=user_buffer,-
        usz=512

user_buffer:
    .blkb 512

record_buffer:
    .blkb 512

fiel_name:
    .ascii /SYS$DISK:[OSCAR]READ.MAR/
file_size=-.file_name

.psect text
orlf: byte 13,10
quote: .byte 39
eofmsg: .ascii /[eof]/<13><10>
eoflen =.eofmsg

;
;main code starts here
;
.psect code
read: .word 0
    $assign_s,-
    devnam=tt,-
    chan=ttchan
    blbs r0,5$
    ret
;open the current terminal
;open worked
;open failed

;
;now open the file
;
5$: $open fab=fab_block
    blbs r0,10$
    ret
;open worked
;open failed

;
;connect to the file
;
10$: $connect,-
    rab=rab_block
    blbs r0,20$
    ret
;connect worked
;connect failed

;
;get next record
;
20$: $get rab=rab_block
    cml r0,$rms$eof
    bneq 30$
    brw 90$
;watch for end of file
;not yet
;got eof

30$: blbs r0,40$
    ret
;watch for other errors
;something else is wrong; quit

40$: movl rab_block+rab$w_rsz,r2
    print quote,$1
    print user_buffer,r2
    print quote,$1
    print orlf,$2
    brw 20$
;get length of record just read
;output a '"'
;output the record just read
;output another '"'
;output a <CR><LF>
;loop till end file file

90$: print eofmsg,$eoflen
    $close fab=fab_block
    ret
;say we're at end of file
;close the file
;exit to vms

.end read

```


TIPS & TECHNIQUES

A Column For The Advanced RSTS/E User

Wef Flesichman, Software Techniques, Inc.

RSTS JOB CONTEXT

1.0 INTRODUCTION

In this column we shall explore RSTS job context: what the term means, why RSTS uses it, where it keeps it, and how your understanding of it can help you in debugging MACRO programs. To introduce you to the context concept we will first turn our attention to various types of information which describe jobs on the RSTS system (more than that which you see in a SYSTAT listing).

2.0 JOB DATA STRUCTURES

Many of you are familiar with the job data structures, the "JDB," the "JDB2," the "I/O block" and the "work block." These data structures hold important job information and they exist for the life of every RSTS job, from HELLO to BYE. This data is used relatively often by the monitor, both during the active execution of the job and also when other, seemingly unrelated, situations arise. Because these structures are so often accessed, and on so many diverse occasions, they are kept in that precious system memory area known as the small buffer pool.

In contrast, other job information is only needed when it is the job's turn to execute. In this case it is more viable to store such information directly in the job's memory image, which will swap out of memory along with the rest of the job when not needed. Included in this category are, for example, the control-C transfer address, the job's general processor registers R0 through R5, the stack pointer, and the program counter. This information is kept in the first 1000 octal bytes of the job image, and thus it is called the job's "lowcore."

The job lowcore area is critical to run-time systems as well as the RSTS monitor because it provides a convenient read/write data area to keep job-specific context in the form of status words and pointers. This design has been critical to the successful emulation of other PDP-11 operating systems, such as RT11 and RSX-11M, whose context information varies slightly from the interpretation given by RSTS.

The job data structures and the job lowcore area are both examples of storage areas which the monitor uses for job "context" storage. Context, generally stated, is everything the monitor keeps track of to be able to interrupt a running job and resume it later intact. The ability to store-and-resume-processing is what gives operating systems like RSTS the quality of "time-sharing." Academically, one would say that the current job's context is saved and the next job's context is loaded every time RSTS proceeds to the next user in its round-robin scheduling

scheme. We call this process "context-switching." Context-switching was recognized as such an important function during the design of the VAX-11 processor family, in fact, that major sections of the VAX architecture were significantly influenced.

It should be noted that there are other various places under RSTS where portions of job context are stored as well as those outlined thus far, but we want to focus on the job lowcore area for the present.

3.0 A PRACTICAL APPLICATION

Suppose you wanted to know what some arbitrary job on your system was doing? Say it was a MACRO program you wrote which you believed might be in an infinite loop. What debugging tools would you use to find the problem?

You could rebuild the program with ODT or some other debugging aid and try to observe the aberrant behavior again. But this approach falls short in certain ways: you will very likely wait for a taskbuild, you will have to reinitialize the program which may disturb the error condition, you must hope the debugging aid won't exceed your maximum memory limitation and you must hope the debugging aid doesn't mask the problem you're trying to observe.

Wouldn't it be nice if you could just drop in unobtrusively on the unsuspecting job to sample its current program counter, stack pointer and general registers? If the job were in an infinite loop, this would probably yield substantial information and possibly allow the isolation of the problem to a defective module.

The accompanying program, JOBCTX, displays the last-stored register context of any job on a RSTS system. It happens that the monitor stores this information in locations 62-103 of each job's lowcore. This information is readily accessible if the job is resident in memory, because we can simply peek at the job using the XPEEK UUO. If the job has swapped out, then we can retrieve the same information from the swapfile in the JDB-indicated swap slot. We can be sure that the target job's context is "frozen" at the time we look at it because its context had to be saved in order for the JOBCTX program to execute. However, we must realize that JOBCTX could be interrupted during its own execution and the target job could resume. This we can detect by monitoring various target job parameters before and after we look at it. Other events may occur as well. The target job may swap in or out of a swapping slot, it may move to another area of memory or it may be killed. If context is invalid for the job, then JOBCTX attempts a retry. If for some reason, 127 successive retries fail, the program reports that the "%Job is elusive." A single retry will usually be sufficient to capture accurate data, but on a heavily loaded system numerous retries are certainly fathomable.

An extension of JOBCTX could easily print out the current FIRQB and XRB for the target job as well as the general registers. The author has written a program to create dump images of jobs which can be analyzed by the DEC PMDUMP utility. One might contemplate a change to SYSTAT to embellish its job reporting statistics.

Don't get caught out of context! Good luck.

**In over 50 installations,
on four continents...**

**RSTS System Managers
will sleep tonight.**

LOCK-11

provides them with:

- **Comprehensive access control**
(35 machine years without a verified breach)
- **Powerful system management tools**
(that don't degrade the system they manage)

Circle the number below for a full set of documentation to evaluate.

CIRCLE 12 ON READER CARD

**RSTS
SECURITY**

'against all enemies,
foreign and domestic'

LOCK-11

NDD 70 James Way
Southampton, PA 18966
(215) 364-2800

IT'S 2:28 AM

Some kid with a MODEM just figured
out that you have 1,100, DEMO.

LOTS OF LUCK!

LOCK-11

**IT'S 3:15 PM
MONDAY**

Tired of writing depreciation journals in
3.5.GL, your third assistant bookkeeper
just discovered the joys of 4.0.PAY.
He's on his way from the bank
to the airport.

LOTS OF LUCK!

LOCK-11

IT'S 2:28 AM

The kid with his auto-dial MODEM
just found your "new" dial-in number
555-0112 on the 112th try.
He's in and you are out.

LOTS OF LUCK!

LOCK-11

**IT'S 5:30 PM,
FRIDAY**

Your FORMER programmer just went home.
He dialed into a non-priv account, let himself
in through a "back door", (1, 82) X TSK (232)
He is now linking the bottom of (1, 2) to the
top with ODT. He is planning a couple of
custom monitor patches.
He is not mad anymore...

LOTS OF LUCK!

LOCK-11

NDD 70 James Way, Southampton, PA 18966
(215) 364-2800

[illegible]

```

CALL      PRINT,R5,LBLTXT ;PRINT REGISTER LABELS
MOV       #BUFF+MONCTX+2,R4 ;POINT TO SAVED R0 CONTEXT VALUE
MOV       #9.,R5           ;SET NUMBER OF REGISTERS TO PRINT
MOV       #-1,R2          ;SPECIFY LEADING ZEROES

20$:      MOV       (R4)+,R1 ;GET NEXT REGISTER
MOV       #BUFF,R0        ;PLACE TEXT IN BUFFER
CALLX     $CBOMG          ;CREATE PRINT STRING
MOVB      #BLANK,(R0)+    ;SEPARATE NUMBERS BY A BLANK
CLRB      (R0)            ;TERMINATE RETURNED STRING
CALL      PRINT,R5,BUFF    ;PRINT
SOB       R5,20$          ;BACK TILL ALL PRINTED
MSGAG     <<CR><LF>>      ;TIDY UP CARRIAGE

        MOVVB      RETRYS,R1 ;DID DUMP REQUIRE RETRIES?
        BEQ        40$      ;NO, WE'RE REALLY DONE TH^C

?Program aborted

.SBTTL    SAVE TARGET JOB PARAMETERS BEFORE

        .ENABL     LSB

CTXBEF:   MOVB      BUFF+JDMCTL+M.SIZE,SVSIZE ;SAVE SIZE OF JOB IMAGE IN KW
MOV       BUFF+JDMCTL+M.PHYA,SVPHYA ;GET TARGET JOB IMAGE PHYS. ADDRESS
MOV       BUFF+JDDJB2,R3 ;GET POINTER TO JDB2
CALL      GETME3,R5,<40> ;LOAD JDB2 INTO BUFFER
MOV       BUFF+J2KCT,SVJ2KC ;SAVE KILO-CORE-TICKS USED BY JOB TILL NOW
MOV       SVPHYA,R3 ;RETRIEVE STARTING PHYSICAL ADDRESS/100
CLR       R2             ;CLEAR MSB
ASHC      #6,R2          ;CREATE ADDRESS DOUBLEWORD
MOV       #1000,R1       ;GET ALL OF TARGET JOB'S LOWCORE
TSTB      SVJDSW         ;MEMORY OR SWAPFILE MODE?
BEQ       10$            ;TARGET JOB IN MEMORY
CALL      GETSWP         ;GET JOB IMAGE FROM THE SWAPFILE
BR        CTXAFT         ;NOW SEE IF CONTEXT CHANGED

10$:      CALL      GETMEM ;GET JOB IMAGE FROM MAIN MEMORY

.SBTTL    CHECK THAT JOB STATUS DIDN'T CHANGE

        .ENABL     LSB

CTXAFT:   CALL      CHKJDB ;LOAD JDB, JOB RESIDENT?
BEQ       10$           ;YES, DUMP SHOULD BE ACCURATE THEN
TSTB      SVJDSW         ;DUMPING FROM THE SWAPFILE?
BEQ       RETRY          ;NOPE, JOB SWAPPED OUT UNEXPECTEDLY

10$:      CMP       BUFF+JDMCTL+M.PHYA,SVPHYA ;YES, AT SAVED PHYSICAL ADDRESS?
BNE       RETRY          ;NO, BAD NEWS
CMPB      BUFF+JDMCTL+M.SIZE,SVSIZE ;YES, STILL THE SAME SIZE AS BEFORE?
BNE       RETRY          ;NO, BAD NEWS ANYWAY
MOV       BUFF+JDDJB2,R3 ;YES, GET POINTER TO JDB2
CALL      GETME3,R5,<40> ;RETRIEVE JDB2
CMP       BUFF+J2KCT,SVJ2KC ;ANY KILO-CORE TICKS SINCE LAST CHECK?
BNE       RETRY          ;YES, TRY TO CATCH IT WHEN ITS IDLE
CALL      PRINT,R5,LBLTXT ;PRINT REGISTER LABELS
MOV       #BUFF+MONCTX+2,R4 ;POINT TO SAVED R0 CONTEXT VALUE
MOV       #9.,R5         ;SET NUMBER OF REGISTERS TO PRINT
MOV       #-1,R2        ;SPECIFY LEADING ZEROES

20$:      MOV       (R4)+,R1 ;GET NEXT REGISTER
MOV       #BUFF,R0        ;PLACE TEXT IN BUFFER
CALLX     $CBOMG          ;CREATE PRINT STRING
MOVB      #BLANK,(R0)+    ;SEPARATE NUMBERS BY A BLANK
CLRB      (R0)            ;TERMINATE RETURNED STRING
CALL      PRINT,R5,BUFF    ;PRINT
SOB       R5,20$          ;BACK TILL ALL PRINTED
MSGAG     <<CR><LF>>      ;TIDY UP CARRIAGE

        MOVVB      RETRYS,R1 ;DID DUMP REQUIRE RETRIES?
        BEQ        40$      ;NO, WE'RE REALLY DONE THEN
        MOV       #BUFF,R0 ;RETRY COUNT WILL BE POSTED TO THE USER
        CLR       R2       ;SPECIFY ZERO SUPPRESSION, LEFT JUSTIFY
        CALLX     $CBDMG    ;CREATE COUNT AS A DIGIT STRING
        CLRB      (R0)     ;TERMINATE STRING
        MSGAG     <<"Required">
        CALL      PRINT,R5,BUFF ;PRINT THE NUMBER OF RETRIES
        BIT       #^C<1>,R1 ;MORE THAN ONE RETRY?
        BEQ       30$      ;NO
        MSGAG     <<" retries"<CR><LF>>
        BR        40$      ;EXIT
        MSGAG     <<" retry"<CR><LF>>

30$:      MSGAG     <<">

40$:      MOV       #FIROB+QFQUN,R0 ;POINT TO FIROB @ QFQUN
        CLRB      (R0)+    ;INDICATE CLOSE FUNCTION
        CLSFO     EQ,0     ;ELSE THE ABOVE WON'T WORK
.ASSUME   MOVB      #SWPCHN,(R0) ;DECLARE CHANNEL TO CLOSE
        CALFIP    ;CLOSE THE SWAPFILE CHANNEL (IF OPEN)
        CALL      ERRCHK   ;ERRORS?
        JMP       RESTRT   ;THAT'S ALL FOR NOW

RETRY:    CALL      CLENUP  ;CLEAN UP FROM AN UNUSUAL SITUATION
        INCB      RETRYS   ;INCREMENT RETRY COUNT (MAX RETRIES = 127)
        BMI       50$      ;RETRIES EXHAUSTED, GIVE UP
        JMP       GO       ;RETRIES REMAINING, RETRY THE DUMP

50$:      MSGAG     <<"%Job is elusive"<CR><LF>>
        JMP       RESTRT   ;RESTART

.SBTTL    CHKJDB - LOAD JDB AND CHECK FOR RESIDENCY
;+
; CHKJDB - READ JDB INTO BUFFER AND SET "Z" BIT IF JOB IS RESIDENT
;
; CALL      CHKJDB
;
; JDB LOADED IN "BUFF"
; Z = 1 IF JOB IS RESIDENT
; ALL OTHER REGISTERS PRESERVED
;-

        .ENABL     LSB

CHKJDB:   MOV       R2,-(SP) ;SAVE REGISTER(S)
MOV       R3,-(SP) ;
MOV       JDBADR,R3 ;GET POINTER TO JDB
CALL      GETME3,R5,<40> ;RETRIEVE THE JDB
MOV       (SP)+,R3 ;RESTORE REGISTER(S)
MOV       (SP)+,R2 ;

```

CIRCLE 11 ON READER CARD

RSTS/E INTERNALS MANUAL

The RSTS community has been clamoring for years for a book that details the inner workings of RSTS/E. Well, clamor no more. Michael Mayfield of Northwest Digital Software, and M Systems, the publisher of The RSTS Professional and The DEC Professional Magazines, have teamed up to produce the RSTS/E Monitor Internals Manual.

This manual describes the internal workings and data structures of the RSTS/E monitor. It also notes differences in the internal structures between version 7.1 and earlier versions of the monitor. Future updates will include changes for new versions of the monitor.

Information is available for all levels of users:

- Gain a basic understanding of the workings of the monitor for optimizing system performance.
- Information on disk structures allows recovery of data from corrupted disk packs.
- Special uses of runtime systems and resident libraries allow complex applications to be developed without degrading system performance.
- Write your own custom device drivers for that "foreign" device you need to add but thought you couldn't.

CONTENTS:

Chapter 1 describes the structures used by the monitor that are resident on disk. These include the directory structure, disk allocation tables, Save Image Library (SIL) formats, bootstrap formats and bad block mapping.

Chapter 2 describes the tables used within the monitor to control system resources and provide program services. These tables provide job, memory, file and device control, as well as program services such as interjob communication.

Chapter 3 contains information on writing and installing a custom device driver. It describes the entry points and information the driver must provide to the monitor as well as the subroutines and macros the monitor provides for the driver.

Chapter 4 contains information that enhances information already provided by Digital on writing custom resident libraries and runtime systems. It concentrates mainly on non-standard uses of resident libraries and runtime systems to increase system performance and functionality.

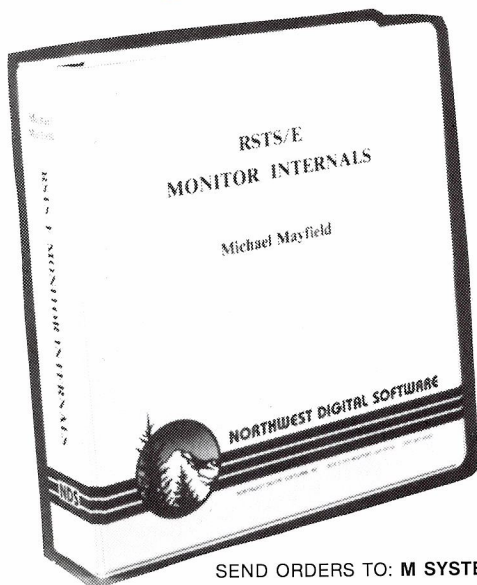
Appendix A provides six quick reference foldout charts:

- The directory structure.
- The monitor tables.
- Fixed memory locations and common data structures.
- Monitor subroutines.
- Device driver entry points.
- Device driver macros.

Appendix B provides examples of the peek sequences required to access most of the monitor tables. It also contains an example program that uses many of the monitor tables to display a job and open files status.

Appendix C provides an example device driver.

Appendix D provides an example runtime system that doubles as a menu system for restricting specified users to a menu of options.



\$95⁰⁰

SEND ORDERS TO: M SYSTEMS, INC., BOX 361, FORT WASHINGTON, PA 19034-0361

Yes Virginia, there is a Macro Man!

Call MACRO MAN

for superior performance

- RSTS Internals •
- Custom Macro Programming •
- RSTS or 11M •

LEARN MACRO FROM THE MASTER!
Seminars available — call for details.

MACRO UTILITY LIBRARY
Available now — call for information.

Also Available Now —
VAX MACRO Consulting

NEXT RSTS MACRO COURSE MID JANUARY 1983

Bob 'Macro Man' Meyer
9 Lockwood Avenue, Fieldsboro, NJ 08505
609-298-9127

CIRCLE 84 ON READER CARD

Going from RSTS/E to VAX/VMS?

MIGRAID3

can help get you there.

MIGRAID3™ USERS WITH A RSTS/E DISK PACK MOUNTED AS A "FOREIGN" VOLUME UNDER VMS CAN

- TRANSFER RMS-11 FILES **DIRECTLY** TO VMS, READY TO USE
- COPY ASCII STREAM FILES **DIRECTLY** TO RMS-32 SEQUENTIAL/VARIABLE FORMAT
- TRANSFER VIRTUAL ARRAYS AND BLOCK I/O FILES, READY TO **OPEN ORGANIZATION VIRTUAL** (SUPPORTS ROSS/IV FILE STRUCTURES)

MIGRAID3 SUPPORTS RSTS/E WILDCARDS FOR BULK COPIES, AND OFFERS A COMPREHENSIVE DIRECTORY FACILITY TOO!

\$800 (U.S.) Single-CPU license;
QDA/OEM terms available.



COMPUTER METHODS CORPORATION
P.O. BOX 592
MOORESTOWN, NJ 08057
(609) 778-8440

RSTS and VAX/VMS are registered trademarks of Digital Equipment Corporation.
ROSS/IV is a trademark of Evans, Griffiths and Hart, Inc.

CIRCLE 152 ON READER CARD

HOW TECO

By Andrew Harrison, Schroder Life Assurance Ltd.,
Enterprise House, Portsmouth, England

I read the article on getting directories in TECO (RSTS Professional, June, 1981) & noticed mention of the problems with wild card edits. The attached TECO macro is designed to allow correct wild card editing. It can either be installed as an initialiser or run from TECO. I trust that modification for other than PDP11/70 systems will prove trivial.

If it is used as an initialiser then it will only work on the simplest format & will return control to TECO in all other cases. If it is run from TECO it expects the buffer to contain the filespec. The default extension is .BAS (but others could be used). (Anyone who puts the account after the file name "FRED[1,2]" deserves the error message he gets; however, if required, the routine to insert ".BAS" can be altered to:

```
OJ :S.$ "E HXP EN!EQP$ HK G* OJ S.$ IBAS$'
```

this used the ENfilespec\$ to put it in standard format.) If no device is specified then after trying the default disc an alternative is tried so the routine !ADD C:! **MUST** be altered to a logical that is meaningful for your system (or removed). (Masochists can extend this routine to scan several discs.) For efficiency the wild card routine is skipped if the filespec does not contain wild cards (it saves at least one directory look up).

After running the routine there are 3 commands provided:

MF\$\$ finishes the edit and reedits leaving you at the front of the file

MN\$\$ finishes this edit and edits the next file in the wild card specification

-IMN\$\$ kills the current edit and then edits the next file in the wild card specification

The method used is to store all the matching file specs in Q register 0 separated by <cr><lf>. This freezes the list of files and prevents the wild card lookup getting confused. File names are then stripped off the front of this list & used (:GO\$\$ can be used to inspect the list of files not yet edited.)

All the "\$" signs are ESCapes, the "!" symbol can usually be used directly but some of the "!" must be genuine.

```
! TECO initialiser routine!
! buffer contains the CCL this routine only handles the simple format!
! TE FILESPEC or TECO FILESPEC!
! alternatively as a program load the buffer with the FILESPEC!
! To test use EI, but for running use the SQUished version!
```

```
! Q register usage!
! N edit next wild card file!
! O work file for above!
! F find front of file!
! P file name for above!
```

```
! first step is to jump to end if CCL contains "/" or "="!
OJ :S/$ "N OEND$!" :S=$ "N OEND$!"
```

```
! delete "TECO" from buffer to use buffer for formatting instructions!
! do this by finding space & deleting up to it!
OJ :S $ 0,.K
```

```
! now add default extension of .BAS if none given!
OJ :S.$ "E ZJ I.BAS$'
```

!OPEN!


```

HXP      !copy file spec to register P!
! now check for either * or ? which indicates a wild card file spec!
OJ :S?*"N OWILDS$' :S*?"N OWILDS$'

! here if no wild card found so do simple edit!
:EB$^EQP% "E OADD C:$' ! do EB of register P, if failure then try
adding C:

HK G* HXP      !file opened name in P!
^AEditing file "A HT ^A"
^A HK Y
OUTILITIES$

!WILD!

@EN% ^EQP%      ! set file spec in wild card buffer!

< :EN$ ;      !look up file spec!
HK G* I
$ :HXO > !append filesec <cr><lf> to 0 register !
HK GP OJ :S:$ "E OADD C:$'

! next load N with a find & edit next wild filename command!
@^UN\
+O "L EK HK ^Aedit killed
^A ! EC :QP "N ^Aedit finished
^A''      ! either abort or finish edit!
! print "edit finished" only if there is something!
! (i.e. file name) in P!

GO OJ :S
$ "N      !put unused file specs in buffer & find end of next file spec!
..ZXO      !save unused file specs!
O,.-2XP ! put current file spec into P!
^Aediting file "A O,.-2T ^A"
^A @EB$^EQP% !do EB on contents of register P!
HK Y ! ^A*** No more files ***
^A '\
:QO "E OADD C:$' ! this gives file not found message if 0 is empty!
HK XP MN ! use the routine!
OUTILITIES$

!ADD C:!! here if file not found so if no disc specified try C:
OJ :S:$ "E IC:$ OOPEN$'
^AFile not found - sorry
^A HK

!UTILITIES!! load utilities into Q registers!.

! next add a find front of file routine if P!
@^UF^EC^Afinish edit
^A :QP "E G* HXP HK' ! load P with G* if P was empty!
@EB$^EQP%      ! do EB of register P!
^Are edit "A :GP ^A"
^A Y\

^Ainitialiser has run
^A
^C$ ! this unconditionally ends macro, if this is an initialiser!
! then it jumps straight into TECO rather than via TECO.TEC!
!END!
$$

```

Naturally people can add routines to load their own useful utilities in the utilities section.

Some of the routines I would like to develop are to do with the TECO screen handling routines. If someone who understands the ":W" commands could write an intelligible article on their use I would be both amazed and grateful. In particular is it possible to get a "SEEALL" mode display without loading "VTEDIT" ?

A more recent copy of RSTS Professional gave some interesting details of EDT initialisers, etc. This has led me to some interesting experiments on both RSTS and VAX. Here are some random jottings on the minor differences between the two implementations of the EDT screen editor. Special characters such as <ESC> and <FF> are handled by displaying in the above formats, the VAX EDT (in my opinion correctly) leaves them embedded in the text:-

"line with <ESC> inserted in it"

but RSTS EDT treats them as line terminators:-

"line with <ESC>

causing a new line."

fault tolerant PDP-11 systems

(n.) Something that prevents perfection, an error.

(adj.) Able to withstand or endure an adverse environmental condition.

(TM) Popular minicomputer family from Digital Equipment Corporation.

(n., pl.) Groups of interacting, interdependent elements forming collective entities.

(Product) 1. Redundant, multiprocessor configurations of DEC standard hardware enhanced by advanced components. 2. Compatible habitat for DEC software products. Ibid. VAX.



(Supplier) 1. The source of such systems, components, and software. 2. A systems house, designing such products to order.

WESTERN RESOURCE TECHNOLOGY, INC.
2970-Q East La Palma, Anaheim, CA 92806
Telephone (714) 630-7852 TWX (910) 591-1241

reliable by definition

CIRCLE 167 ON READER CARD

Minitab
Minitab
Minitab
Minitab
Minitab
Minitab
Minitab
Minitab
Minitab
Minitab
Minitab™

The Only Interactive Data Analysis Package for All DEC Systems

Elementary Operation, Advanced Capability

Used by:

- One third of *Fortune's* Top 50
- Almost every major university in the U.S.
- Hundreds of smaller organizations in 25 countries around the world.

Its simplicity of design has been steadily perfected since 1972. The results are:

- Unusually rapid installation and operation
- A sophisticated range of data analysis capabilities and statistical procedures.

**You can do more
when you do it simply.**

For more information write or phone: **Minitab Project**,
215 Pond Laboratory,
University Park, PA 16802 U.S.A.
Telephone (814) 865-1595
Telex 84-2510

CIRCLE 168 ON READER CARD

Carico Center
2845 NW 62nd Street
Ft. Lauderdale, Florida 33309
Telephone (305) 979-2844

CIRCLE 49 ON READER CARD

Finally some enhancements that I would like to see. The advance word key skips multiple spaces but does not skip multiples of any other delimiter; I would like it to do so. It is frustrating and perplexing the way EDT slows down the repeat rate on the keypad keys, how does it do this ? and why? Screen width, EDT ignores both the actual terminal setting and also the width set on the system; surely it could pick up one or the other. There is no facility to return to the point marked by using the SELECT key “.”. ❤

... continued on page 66

NOTE: The following article is based on a paper presented at the Spring-82 DECUS and included in the DECUS Proceedings.

A USER WRITTEN KEYBOARD MONITOR

By Ken Harris, Unico, Inc., 3725 Nicholson Road, Franksville, WI. 53126, (414) 886-5678

INTRODUCTION

This article describes the implementation of a keyboard monitor designed to provide RSTS/E users with a more powerful command environment. The backbone of the keyboard monitor is a macro processing facility which allows users to define their own commands and symbols. The intent of this article is to describe the interesting details of this project and to provide some ideas for those interested in similar efforts.

WHAT IS A KEYBOARD MONITOR

On a RSTS/E system, users communicate with the operating system via command interpreters called keyboard monitors. A keyboard monitor prompts the user for input, waits for a command, and then translates the command into some action. That action may be running a program, adding a private logical, clearing a crt screen or whatever. It is the keyboard monitor that defines the command environment that the user sees. DIGITAL supplied keyboard monitors include DCL, BASIC-PLUS, RSX and RT11.

WHY BUILD A KEYBOARD MONITOR

Different users have different needs and desires. The needs of a programmer are certainly different from those of an accounting clerk. By writing a keyboard monitor, the user interface may be tailored to meet specific needs. The command set can be expanded, restricted or made more friendly. Support can be provided for specific terminal devices.

HOW TO BUILD A KEYBOARD MONITOR

A keyboard monitor may be implemented in a number of ways. We will restrict our attention to two approaches which for our purposes will be referred to as the High-road and Low-road approaches.

The High road approach involves the implementation of a keyboard monitor as a run time system. This is not an easy task. It requires a knowledge of runtime systems and Macro-11 programming. The advantage of this approach is an efficient product which should be fast, compact and shared. The disadvantage is the difficulty involved with

BAC_{mac} can do it all!

BAC into RTS / BAC into MAC / BAC into BAS

**Two Word Version
Now Available**

BACmac is a unique software tool, running under RSTS/E, which provides the following conversions:

- translation from Basic-Plus "compiled" back to Basic-Plus source code (only the comments will be missing)
- translation from Basic-Plus into Macro source code, which compiled under RSTS runs faster than Basic-Plus
- translation from Basic-Plus into Macro source code which may be compiled under RSTS for execution under RT11 — a migration facility
- translation from Basic-Plus into a RUN-TIME-SYSTEM. Now you can write an RTS in Basic-Plus. The ideal solution to memory thrashing due to "multi-copy" applications programs.

RSTS/E, RT11, Macro-11 and Basic-Plus are trademarks of Digital Equipment Corporation.

Western Distributor:
Telecom Computer Systems, Inc.
P.O. Box 03285
Portland, Oregon 97203
503/286-5122

ADOS | Advanced Digital
Office Systems

Eastern Distributor:
New England Micro Technology, Inc.
P.O. Box 767
Marblehead, Mass. 01945
617/631-6005

The Low-road approach involves the use of feature patch 21.3.5 to create a modified RSX runtime system that chains to the keyboard monitor program. The keyboard monitor may then be written in any available language. The advantage of this approach is the ease of implementation. The disadvantage is that the keyboard monitor will not be as efficient as a runtime system.

The remainder of this article is devoted to a discussion of a keyboard monitor called PDM. PDM (or Program Development Monitor) is a Low-road monitor written entirely in "C". It was designed and implemented here at UNICO and has been in use for about a year.

The heart of PDM is a macro processing facility which allows users to define their own commands and symbols.

(1) To provide a command environment which can be easily tailored to the needs of a specific task.

(2) To provide a command environment which promotes programmer productivity.

(3) To provide a command environment which enhances rather than replaces the existin

(4) To start with a small keyboard monitor and add enhancements as needs and ideas arise.

(5) To provide a general tool.

The following is a general description of what PDM does with a command.

(2) **Get Command Line** — PDM prompts the user for input with a colon ":", inputs a line of text and converts it to upper case.

These are commands which change or display PDM's context information and must be executed before macro expansion occurs. If the input command is a first level command, it is executed and control returns to get the next command.

(4) **Macro Expansion** — Next the command is passed through the macro processor. If the command expands into more than one line, it is forced to the job's console and control returns to get next command.

(5) **Second Level Commands** — PDM next checks for a second level command. These are commands which are executed after macro expansion (RUN, PRINT, FORCE, @). If the expanded command is a level two command, it is executed and control returns to get the next command.

(6) Try CCL Command — Since

the command is not a PDM command, PDM tries to execute it as a CCL command. If successful, control is passed to the called program.

(7) **Try DCL Command** — As a last resort, PDM tries to execute the command as a DCL command. Control is passed to DCL. If the command is a DCL command it is executed, otherwise DCL prints an error message and exits to the job's default keyboard monitor, PDM.

FINAR

**The only electronic
spreadsheet that's
powerful, versatile
and easy to use.**

With FINAR it's easy to prepare complex budgets, forecasts, financial statements, cash-flow projections and more. But that's only the beginning. FINAR can perform sophisticated tasks that far surpass any of the "calcs" capabilities. In fact, in every situation where FINAR has been compared to the "calcs", FINAR has been chosen.

FINAR is a real business system, versatile enough to meet the needs of the largest corporations and smaller firms too. FINAR's capabilities include extensive mathematical functions; graphic output; interfaces with other systems; flexible report format; worksheet definition and control directives.



3303 South Rice Avenue, Suite 208
Houston, Texas 77056
(713) 960-0848

CIRCLE 51 ON READER CARD

In just two days one can learn all that is needed to perform the most complex financial analyses with FINAR. Commands are in English—not “computerese”—and simple prompts let one know what information the system requires to compute the answer to a question.

All you need is a DEC PDP-11 with RSTS or a VAX-11 and FINAR to plan calmly for whatever the future has in store. FINAR is available for as little as \$4500. For a demonstration of FINAR's unlimited technical capabilities and ease of operation, call: Walter Fleming (713) 960-0848.

Execute the specified PDM command. If installed as a CCL command, this allows non-PDM users to execute PDM commands.

and their definitions:

To Edit a file "FOO" using the EDT editor:

```
cmd: EDT FOO
```

```
dfn: EDT(FILE) = REDEF "$edfile" = FILE
```

RUN \$EDT

DATA FILE

SET KEYPAD

CHANGE:

The symbol \$EDFILE is defined

for later use by the re-edit

last file function key PF4.

To Build a Basic+2 program "FOO":

cmd: BP2 F00

```
dfn: BP2-BLD(FILE)=RUN $BASIC2
```

"OLD" FILE

COM/OBJ

BUILD

TKB @FILE

PIP FILE.OBJ.FILE.CMD.FILE.ODL/DE:

ANNOTATED BIBLIOGRAPHY

Brown, P.J., Macro Processors and Techniques for Portable Software. (1974). [This book provides a general introduction to macro processing and includes descriptions of several different macro processors.]

Cole, A.J., **Macro Processors 2nd Edition** (1981).

Kernighan, Brian W. and Plaugher, P.J. "Macro Processing", **Software Tools**, (1976) P251-283. [The macro processor described in this chapter was the starting point for PDM.]

Wasserman,Anthony I.(ED), **Software Development Environments** (1981). [This is an excellent collection of papers covering a wide range of issues relating to software development environments.]

Digital Equipment Corporation, **PDP-11 Macro-11 Language Reference Manual**, (August 1977), DEC AA-5075A-TC.

Digital Equipment Corporation, **VAX/VMS Command Language User's Guide**, (May 1981), DEC AA-D023B-TE.

Digital Equipment Corporation, **VAX/VMS Guide to Using Command Procedures**, (March 1980), DEC AA-H782A-TE.

Digital Equipment Corporation, "Notes on Writing Keyboard Monitors", **RSTS/E V7.2 Maintenance Notebook**, (June 1982), Seq 21.1.2 N. [This article contains lots of helpful hints and suggestions.]

Digital Equipment Corporation, "Invoking a Menu Program — RSX.RTS Feature Patch", **RSTS/E V7.2 Maintenance Notebook**, (June 1982), Seq 21.3.5. [This feature patch provides an easy way to implement a keyboard monitor program without making it a run-time system.] ❤️

LOOK at the “tear-out” cards in this issue.

There are subscription cards for you or a friend.

There's a FREE gift for you. Bring in new subscribers and

collect rewards. See "BOUNTY HUNTERS" card.

There's a READERS INQUIRY card for your convenience.

Get the inside story. Get the RSTS Internals Manual.

Get the inside story. Get the RSTS Internals Manual.

CACHE FLOW

By Kevin Paul Herbert, Software Techniques, Inc., 5242 Katella Avenue, Suite #101, Los Alamitos, CA 90720

Copyright © 1982 by Software Techniques, Inc.

Do you have cache flow problems? I thought I understood caching fairly well, until I heard a manager talking about cache flow problems. The managers talk about performance and so do I, but it stops there. I just don't understand how caching relates at all to whether I can get paper clips or fruit juice, but that must just be a mystery of management.

When I think of cache, I think of a fast storage medium (memory) used to store the most frequently accessed portions of a slower storage medium (disk). In many applications, it is possible to cache the most frequently used disk blocks in memory and achieve incredible performance improvements.

I am impatient. Sometimes I am called a speed freak. I don't know. I just want my programs to RUN fast.

I have some knowledge of the RSTS/E directory structure. I decided that I could use this knowledge, combined with my desire for speed, to write some very fast MACRO utilities. I learned a great deal about handling cache when working on these utilities, and I now know that it is time to spread that information to the user community. Although the techniques illustrated evolved out of use of MACRO-11 and RSTS/E, they are appropriate to all languages.

I refer to items in my cache as cache entries. A cache entry consists of a cache header followed by the cache data. Throughout all implementations of the cacher, I used contiguous address space within my job for the cache.

A cache stores only a portion of the data from the secondary storage medium. A cache entry will be removed from the cache only when the memory is needed to cache newer data. This is referred to as a cache replacement.

A cacher can either be read-only or read-write. One of my utilities required a read-only cache, and the other required a read-write cache. The read-write cache was implemented using deferred writes. This means that when I made a change to the data stored within a cache entry, I set a flag in the cache header. When the cacher needs to replace an entry, it sees that the written-to flag is set, and then writes the date out to disk. This way, disk activity occurs only at the last possible moment.

I was very interested in determining the effectiveness of the cacher, so I included statistic gathering code in all implementations. The information that I kept was the number of cache replacements, the number of accesses to the cache, and the number of accesses by the cacher to disk. These statistics are invaluable in determining an appropriate implementation.

The cache header used in my first implementation is structured as follows:

1. OLDNESS.

The OLDNESS element is set to zero when data is first installed in the cache. This counter is incremented every time a new entry is installed in the cache. This counter is decremented (to a minimum of zero) every time this cache entry is accessed. (In some applications, it is possible to reduce the number of cache replacements by zeroing the counter on access instead of decrementing it.) When it becomes necessary to perform a cache replacement, the cacher scans all the cache headers to find the cache entry with the largest OLDNESS.

2. BLOCK

The BLOCK element contains the disk block number associated with this cache entry. When the cacher is called to locate a disk block, it scans all the cache headers to find a matching block number before performing a cache replacement and reading the block from disk.

All scanning of cache headers is performed through sequential access. Disk activity is reduced a great deal through this implementation, but there is a high cost in CPU time.

My second implementation addressed the CPU time issue. After examining the statistics I had collected, I concluded that the newer a cache entry was, the more likely it was to be accessed. I therefore directed my energy toward reducing the number of memory accesses necessary to locate a cache entry. The cache header used in the second cacher is set up as follows:

1. LINK-TO-NEXT-OLDEST

The LINK-TO-NEXT-OLDEST element contains the address of the next oldest cache entry. When an entry is added to the cache, a cell called LINK-TO-THE-NEWEST is set to contain the address of this new entry, and the LINK-TO-NEXT-OLDEST element is set to the value previously stored in LINK-TO-THE-NEWEST. Whenever this cache entry is accessed, it is replaced as the LINK-TO-THE-NEWEST entry since it is the likeliest cache entry to be accessed upon the next cache access.



CIRCLE 134 ON READER CARD

2. LINK-TO-NEXT-NEWEST

The LINK-TO-NEXT-NEWEST is used as a backwards link to implement quick cache replacements. A cell LINK-TO-THE-OLDEST is used to point to the oldest cache entry. When a cache replacement occurs, the value of LINK-TO-NEXT-NEWEST is written to LINK-TO-THE-OLDEST.

3. BLOCK

The BLOCK element contains the disk block number associated with this cache entry. When the cacher is called to locate a disk block, it scans all the cache headers to find a matching block number before performing a cache replacement and reading the block from disk.

This implementation reduced CPU time used by the cacher without affecting the disk activity at all. There were two areas of improvement in the cacher. First, I reduced CPU time necessary to locate cache entries. This occurred because of my statistical observation dealing with the way I accessed the disk. Second, it was no longer necessary to scan all the cache blocks to find the oldest entry. This improved performance when there were many cache replacements.

- **COLINK**, a convenient, efficient package for using DMCIs or DMRIIs without the overhead of file transfers, virtual terminals, and other features.
- **DIALUP**, a comprehensive, efficient package for systems using asynchronous terminal connections, virtual terminals, auto-dialing, and other features. The premier RSTS/E package for efficient file transfers.

DEC, DECnet, RSTS, RSX, VAX, and VMS are trademarks of Digital Equipment Corporation.

Call or write for complete description.

Evans Griffiths & Hart,
55 Waltham Street, Lexington, MA 02170
(617) 861-0670

I was not yet satisfied. Some person at DECUS dared me to make my program run even faster. I consulted Deep Thought and came up with an answer. The newest and oldest cache entries were now locatable just by examining

one fixed cell. That improved performance a great deal. What if I could locate any cache entry by just examining one cell (rather than scanning the block entry in the cache header)? This would reduce CPU time even further!

I concluded that if the range of block numbers was

relatively small, I could have a cell for each possible block number. This cell would either contain a zero to indicate that the block was not in the cache, or contain the address of the cache entry. One of my utilities used the cache to cache an entire RSTS/E file structured disk. There was no way to have a cell for each block of an RM05, so I forgot the idea in that utility. The other utility, however, cached a RSTS/E UFD and a temporary work file. The total number of cells that I would need in this utility was only 14. This was very workable.

It was not necessary to change anything in the cache header. The block number was still necessary to facilitate cache replacements. All I added was a static region containing 14 cells, one for each possible cache entry. This change made a noticeable

increase in performance and finally satisfied me.

Who says speed kills? I program for speed, and I hope that reading this article has excited the speed freak in you.

Have fun, and don't have any cache flow problems! ❤️

Software Tools for RSTS/E

Evans Griffiths & Hart, Inc., a pioneer in the development of RSTS and the winner of an ICP million dollar award for KDSS and TAM, offers packages that save you time and improve your productivity.

- **KDSS**, a complete multi-terminal key-to-disk data entry subsystem. Eliminates the need for keypunching and stand-alone key-to-disk systems. (Also available for VAX/VMS and RSX-11M.)
- **TAM**, an efficient multi-terminal screen-handling facility that provides complete support for the development of transaction-processing applications on a wide variety of terminals. (Also available for VAX/VMS and RSX-11M.)
- **FSORT3**, a very fast sort/merge package for RMS and non-RMS files. More economical of disk space than SORT-11 and much faster.
- **SELECT**, a convenient, very fast package for scanning files to extract records that meet user-specified selection criteria. Use as part of an online inquiry system and as a front end for building file indices and generating reports. SELECT and FSORT3 can save hours in nightly batch runs.
- **BSC/DV**, a RSTS/E device driver for the DEC DV11 synchronous multiplexer. Suitable for handling a wide variety of bisynchronous protocols. (Also available for VAX/VMS.)
- **COLINK**, a convenient, efficient link between two RSTS/E systems using DMC11s or DMR11s without the overhead of DECnet. Supports file transfers, virtual terminals, and across-the-link task communication.
- **DIALUP**, a comprehensive, efficient link between RSTS/E and other systems using asynchronous terminal lines. Supports file transfers, virtual terminals, auto-dialing, and the use of command files and macros. The premier RSTS/E package for remote support and reliable, CPU-efficient file transfers.

DEC, DECnet, RSTS, RSX, VAX, and VMS are trademarks of Digital Equipment Corporation.

Call or write for complete descriptions of features and benefits.

Evans Griffiths & Hart, Inc.
55 Waltham Street, Lexington, MA 02173
(617) 861-0670

EGH

[illegible][illegible][illegible]

Line	Code	Comment	Line	Code	Comment
	ELSE		3700	!	DISPLAY MDR BIT PATTERN
	T50\$ = " " + T50\$		3710	Z9% = FNCURSOR\$(8%,32%)	
1610	FNCVT\$ = T50\$		3720	Z9% = FNP\$(MDR\$)	
1620	FNEND		3730	RETURN	
1630	!		3740	!	
2000	MAR% = AD% : GOSUB 3100		3800	!	DISPLAY SP BIT PATTERN
	\MDR% = ST\$(MAR%) : GOSUB 3700		3810	Z9% = FNCURSOR\$(10%,15%)	
	\RETURN		3820	Z9% = FNP\$(SP%)	
			3830	RETURN	
2100	MAR% = SP% : GOSUB 3100		3840	!	
	\MDR% = ST\$(MAR%) : GOSUB 3700		3900	!	DISPLAY X BIT PATTERN
	\RETURN		3910	Z9% = FNCURSOR\$(10%,38%)	
			3920	Z9% = FNP\$(X%)	
2200	T20%=0%		3930	RETURN	
2210	IF (ACC% AND MDR%) < 0%		3940	!	
	THEN T20% = -1%		3950	!	INDIRECT ADDRESSING
	ELSE IF (ACC% OR MDR%) > 0%		4000	!	
	THEN T20% = +1%		4010	MAR% = AD% : GOSUB 3100	
2220	ACC% = ACC% + MDR% : GOSUB 3300	! ADD	4020	MDR% = ST\$(MAR%) : GOSUB 3700	
2230	IF (ACC% XOR T20%) < 0%		4030	AD% = MDR%	
	THEN OV% = 1%		4040	RETURN	
	ELSE OV% = 0%	! OVERFLOW ?	4050	!	
2240	GOSUB 3500		4099	!	PRINT INSTRUCTION
2250	RETURN		4100	Z9%=FNCURSOR\$(14%, 44%)	
2260	!		4120	PRINT ELINE\$;	
2300	MDR% = -MDR%		4130	PRINT FNOCT\$(PC%-1%,ADL%) + " ";	
2310	GOSUB 2200		4140	PRINT FNPI\$(IR%);	
2320	RETURN		4150	RETURN	
2330	!		4200	!	PRINT MESSAGE
2400	T20=ACC% *MDR%		4210	Z9\$ = FNPRINT\$(ERRM\$)	
2410	IF T20>32767 OR T20<-32768		4220	Z9% = FNLINEFEED%	
	THEN OV%=1%		4230	RETURN	
	ELSE		4240	!	
	OV%=0%		4300	!	SYNTAX ERROR
2420	GOSUB 3500		4310	ERRM\$ = "SYNTAX ERROR " + NUM\$(FLAG\$)	
2430	ACC% =T20 : GOSUB 3300		4320	GOSUB 4200	
2440	RETURN		4330	!	
2450	!		4340	RETURN	
2500	T20% = ACC% / MDR%		4350	!	
2510	R% = ACC% - MDR% * T20% : GOSUB 3400		4400	!	I/O FIELD ASSOCIATION
2520	ACC% = T20% : GOSUB 3300		4410	T10% = T10% + 1%	
2530	OV% = 0% : GOSUB 3500		4420	FIELD #1\$, (T10%-1%)*4% AS T11\$,	
2540	RETURN			2% AS T12\$, 2% AS T13\$	
2550	!		4430	RETURN	
2600	MDR% = ACC% : GOSUB 3700			!	
2610	ST\$(MAR%) = MDR% : GOSUB 2900		4990	!	LDA
2620	RETURN		5000	!	
2630	!		5010	GOSUB 2000	
2900	!	PRINT STORE	5030	ACC% = MDR% : GOSUB 3300	
2910	Z9% = FNCURSOR\$(4%,59%)		5040	RETURN	
	\PRINT SPACE\$(9%);		5050	!	
	\Z9% = FNCURSOR\$(4%,59%)		5100	!	STO
	\SLEEP INT((Z1%+2500%)/2500%)		5110	MAR% = AD% : GOSUB 3100	
	\PRINT "S T O R E";		5120	GOSUB 2600	
			5130	RETURN	
2920	RETURN		5200	!	ADD
			5210	GOSUB 2000	
2990	!	DISPLAY PC BIT PATTERN	5220	GOSUB 2200	
3000	!		5230	RETURN	
3010	Z9% = FNCURSOR\$(0%,9%)		5300	!	SUB
3020	Z9% = FNP\$(PC%)		5310	GOSUB 2000	
3030	RETURN		5320	GOSUB 2300	
3040	!		5330	RETURN	
3100	!	DISPLAY MAR BIT PATTERN	5400	!	MUL
3110	Z9% = FNCURSOR\$(0%,32%)		5410	GOSUB 2000	
3120	Z9% = FNP\$(MAR%)		5420	GOSUB 2400	
3130	RETURN		5430	RETURN	
3140					

CXB

```

9990      ! *****
9000      !                                     CLA
9010      ACC% = 0%                               : GOSUB 3300
9020      RETURN
9030      !
9100      !                                     INC
9110      IF ACC% = 32767%
          THEN OV% = 1%
          ELSE
              OV% = 0%
9120      GOSUB 3500
9130      ACC% = ACC% + 1%                         : GOSUB 3300
9140      RETURN
9150      !
9200      !                                     DEC
9210      IF ACC% = -32768
          THEN OV% = 1%
          ELSE
              OV% = 0%
9220      GOSUB 3500
9230      ACC% = ACC% - 1%                         : GOSUB 3300
9240      RETURN
9250      !
9300      !                                     ADDS
9310      GOSUB 12100
          \GOSUB 2100
          \GOSUB 2200
          \GOSUB 2600
          \RETURN

9400      !                                     SUBS
9410      GOSUB 12100
          \GOSUB 2100
          \T20% = MDR%
          \MDR% = ACC%
          \ACC% = T20%
          \GOSUB 2300
          \GOSUB 2600
          \RETURN

9500      !                                     SWAB
9510      ACC% = SWAP(ACC%)                       : GOSUB 3300
9520      RETURN
9530      !
9600      !                                     HOP
9610      PC% = PC% + 1%                         : GOSUB 3000
9620      RETURN
9630      !
9700      !                                     HOPE
9710      IF ACC% = 0%
          THEN GOSUB 9600
9720      RETURN
9730      !
9800      !                                     HOPN
9810      IF ACC% <> 0%
          THEN GOSUB 9600
9820      RETURN
9830      !
9900      !                                     HOPL
9910      IF ACC% <0%
          THEN GOSUB 9600
9920      RETURN
9930      !
10000     !                                     HOPG
10010     IF ACC% > 0%
          THEN GOSUB 9600
10020     RETURN
10030     !
10100     !                                     NOT
10110     ACC% = ( NOT (ACC%))                     : GOSUB 3300
10120     RETURN
10130     !
10200     !                                     NEG
10210     ACC% = -ACC%                             : GOSUB 3300
10220     IF ACC% = -32768
          THEN OV% = 1%
          ELSE
              OV% = 0%
10230     GOSUB 3500
10240     IF ACC% =0%
          THEN C% = 0%
          ELSE
              C% = 1%
10250     GOSUB 3600
10260     RETURN
10270     !
10300     !                                     INCX
10310     X% = X% + 1%                             : GOSUB 3900
10320     RETURN
10330     !
10400     !                                     DECX
10410     X% = X% - 1%                             : GOSUB 3900
10420     RETURN
10430     !
10500     !                                     SWAX
10510     T20% = ACC%
10520     ACC% = X%                               : GOSUB 3300

```



```

11010      IF ACC% < 0%
              THEN T20% = 1%
              ELSE
                  T20% = 0%
11020      ACC% = (ACC% * 2%) OR T20%
11030      GOSUB 3300 :
              IF T20% = 1%
                  THEN T20% = -32768
11040      IF (ACC% XOR T20%)
                  THEN OV% = 1%
              ELSE
                  OV% = 0%
11050      GOSUB 3500
11060      RETURN
11070      !
11100      !
11110      OV% = 0%
11120      RETURN
11130      !
11200      !
11210      OV% = 1%
11220      RETURN
11230      !
11300      !
11310      IF OV% > 0%
              THEN PC% = PC% + 1% : GOSUB 3000
11320      RETURN
11330      !
11400      !
11410      C% = 0%
11420      RETURN
11430      !
11500      !
11510      C% = 1%
11520      RETURN
11530      !
11540      !
11600      !
11610      IF C% <> 0%
              THEN PC% = PC% + 1% : GOSUB 3000
11620      RETURN
11630      !
11700      !
11710      OV% = 0%
11720      C% = 0%

```


[illegible]

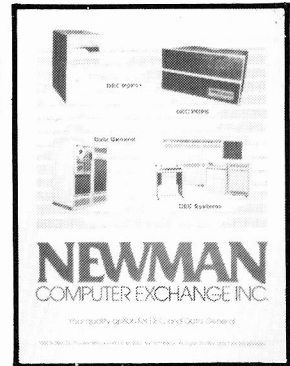
```

17020 IF L9% <> 8%
      THEN ERRM$ = "INVALID RUN COMMAND" :
      Z9$ = FNPRINT$(ERRM$) :
      Z9% = FNLINFEED% :
      GO TO 17770
17030 PC% = FNOCIT$(RIGHT(M$,5%),ADL%)
17040 IF PC% = -1%
      THEN 17700 !NON OCTAL ADDRESS
      ! PRINT PC%
17050 GOSUB 3000
17100 ON ERROR GO TO 32300
17105 Z9$=SYS(CHR$(6%)+CHR$(-7%)) ! SET CTRL/C TRAP
17110 Z9% = FNCURSORS$(14%,0%)
17120 PRINT "RUN MODE - INSTRUCTION BEING OPERATED ON:- ";
17130 PRINT ELINE$ FOR I% = 1% TO 9%
17140 OPP% = 1% : CHP% = 0%
17150 SP% = MEM%+1% : GOSUB 3800
17160 E9% = 0%
17190 !
17200 IF E9% > 0%
      THEN E9% = 0% : GO TO 17700
17205 MAR% = PC% : GOSUB 3100
17210 MDR% = ST$(MAR%) : GOSUB 3700
17220 IR% = MDR% : GOSUB 3200
17230 PC% = PC% + 1% : GOSUB 3000
17240 !
17250 ! SEPARATE OC,AD AND
17260 GOSUB 4100 ! DISPLAY INSTRUCTIONS.
17265 !
17270 IF OC% = 0%
      THEN 17380 ! REGISTER INSTRUCTION
17280 !
17290 IF ( IR% AND MASK$(10%) )
      THEN AD% = AD% + X% ! INDEX MODIFICATION
17300 !
17310 IF ( IR% AND MASK$(9%) )
      THEN GOSUB 4000 ! INDIRECT ADDRESSING
17320 !
17330 !
17340 !
17350 ON OC% GOSUB 5000,5100,5200,5300,5400,5500,
      5600,5700,5800,5900,6000,6100,
      6200,6300,6400,6500,6600,
      6700,6800,6900,7000,7100,7200,
      7300,7400,7500,7600,7700,7800 ! PERFORM INSTRUCTION
17355 GO TO 17260 IF OC% = JSI%
17360 GO TO 17200 ! END OF CYCLE
17370 !
17380 IF AD% = 0%
      THEN 17700 ! "STOP" INSTRUCTION
17390 ON AD% GOSUB 9000,9100,9200,9300,9400,9500,
      9600,9700,9800,9900,10000,10100,
      10200,10300,10400,10500,10600,10700,
      10800,10900,11000,11100,11200,11300,
      11400,11500,11600,11700,11800,11900,
      12000,12100,12200,12300,12400,12500,12600,12700,
      12800,12900 ! PERFORM REGISTER
      INSTRUCTION
17400 GO TO 17200 ! END OF CYCLE
17410 !
17700 Z9% = FNCURSORS$(14%,0%)
17710 PRINT ELINE$;
17720 PRINT "EXECUTIVE MODE"
17730 IF CHP% <> 0% THEN Z9% = FNLINFEED%
17740 !
17760 Z9% = FNCURSORS$(OPP% + 16%, 0%)
17770 RETURN
17780 !
17790 !*****
18000 ! DUM
18001 T11% = 0%
      \T12% = MEM%
      \IF MID(M$,8%,1%) <> "-" THEN
      IF LEN(M$) > 3% THEN
      KB% = VAL(MID(M$,5%,2%))
      ELSE
      KB% = -1%
      ELSE
      T11% = FNOCIT$(MID(M$,5%,3%),ADL%)
      \T12% = FNOCIT$(MID(M$,9%,3%),ADL%)
      \IF LEN(M$) > 11% THEN
      KB% = VAL(MID(M$,13%,2%))
      ELSE
      KB% = -1%
18002 IF KB% > -1% THEN GO TO 18100
      ELSE
      PRINT "PROGRAM " + PRGNAM$ + ".SIM"
      \PRINT
      -
18003 ERRM$ = "ADDRESS CONTENTS"
18004 GOSUB 4200
18005 ERRM$ =
      "OCTAL DECIMAL CHAR BINARY ASSEMBLER"
18006 GOSUB 4200
18009 FOR I10% = T11% TO T12%
18010 IF ST$(I10%) = 0% THEN 18050
18020 T10% = ST$(I10%)
18030 PRINT FNOCIT$(I10%,ADL%), T10%,
      "FNCVT$(T10%) " " FNP1$(T10%) " " FNPI$(T10%);

```

FREE

HIGH DISCOUNT CATALOG OF NEW & USED DEC



SYSTEMS • PROCESSORS • PERIPHERALS

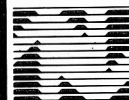
Including

- ★ VAX ★ PDP11
- ★ LSI-11 ★ PDP8
- ★ DECsystem 10 & 20
- ★ Data General, too

NO-RISK TRIPLE WARRANTY

- 1.** Genuine DEC and Data General Equipment
- 2.** 90-Day Warranty on Parts and Labor
- NEW! 3.** 10-DAY RETURN PRIVILEGE. Full credit for any fully-warrantied DEC or Data General item returned within 10 days after receipt . . . and WE'LL PAY RETURN FREIGHT!

Call or write for your copy TODAY!



NEWMAN

COMPUTER EXCHANGE INC.

1250 N. Main, P.O. Box 8610 Ann Arbor, MI 48107

(313) 994-3200

SC22RS



Member
Computer Dealers
& Lessors
Association


```

21000      1
21010      IF L9% > 10%
           THEN ERRM$ = "INVALID UNSAVE COMMAND"
           Z9$ = FNPRINT$(ERRM$)
           Z9% = FNLINFEED%
           GO TO 21080
21020      T10$ = RIGHT(M$,5%)
21030      T10$ = T10$ + ".SIM"
21040      KILL T10$
21050      ERRM$ = T10$ + " SUCCESSFULLY UNSAVED."
21060      GOSUB 4200
21080      RETURN
21090      !
21990      ! *****
21995      ! CATALOGUE
22000      MAT Y% = ZER :
           Y%(23%) = 83% :
           Y%(24%) = 89% :
           Y% = 0%
22010      Y%(0%) = 30% :
           Y%(1%) = 6% :
           Y%(2%) = 15% :
           Y%(4%) = Y%/256%:
           Y%(3%) = Y% - Y%(4%) * 256%
22020      CHANGE Y% TO Y% :
           CHANGE SIS(Y$) TO Y% :
           E$ = RAD$(Y%(11%) + SWAP%(Y%(12%)))
           IF E$ <> "SIM" THEN 22040
22025      ERRM$ = RAD$(Y%(7%) + SWAP%(Y%(8%))) + RAD$(Y%(9%) + SWAP%(Y%(10%)))
22030      ERRM$ = ERRM$ + "." + E$
22032      GOSUB 4200
22034      Y% = Y% + 1%
22040      GO TO 22010
22090      ! *****
23000      ! STORES INPUT INSTRUCTION
23010      ! OR DATA
23050      OC% = 0% ! OP CODE
23060      INDEX% = 0% ! INDEX BIT
23070      INDIR% = 0% ! INDIRECT BIT
23080      AD% = 0% ! ADDRESS
23090      IN% = 0% ! INSTRUCTION
23100      !
23110      STADD% = FNOC$(M1$,ADL%) ! STORE ADDRESS
23120      IF STADD% < 0%
           THEN FLAG% = 1% : GO TO 23760
23140      IF MID(M$,SPPOS%,1%) <> " "
           THEN FLAG% = 2% : GO TO 23760
23160      T10$ = MID(M$,MNEMPOS%,1%)
23170      IF T10$ = " " ! LITERAL CHARACTER
           THEN 23800
23180      IF T10$ = "#"
           THEN IF L9% = OCTLEN%
                THEN 23940
                ELSE FLAG% = 14% : GO TO 23760
23190      IF T10$ < "A"
           THEN 23900 ! DECIMAL INTEGER?
23210      IF T10$ > "Z"
           THEN FLAG% = 5% : GO TO 23760 ! INVALID MNEMONIC
23230      IF LEN(M$) <= REGLEN%
           THEN 23340 ! REGISTER INSTRUCTION
23250      T10% = INSTR(MNEMPOS%,M$," ")
23260      IF T10% = 0%
           THEN FLAG% = 6% : GO TO 23760
23270      ! ! MEMORY REFERENCE
23280      T10$ = MID(M$,MNEMPOS%,T10%-MNEMPOS%)
23290      FOR I10% = 1% TO NOMREF%
23300      IF T10$ = M8$(I10%)
           THEN OC% = I10% : GO TO 23420 ! VALIDATE ADDRESS
23320      NEXT I10%
23330      FLAG% = 5% : GO TO 23760
23340      ! REGISTER INSTRUCTION
23350      T10$ = MID (M$,MNEMPOS%,LEN(M$) - SPPOS%)
23360      FOR I10% = 0% TO NOREG%
23370      IF T10$ = M9$(I10%)
           THEN AD% = I10% : GO TO 23650 !INSTRUCTION COMPLETE
23390      NEXT I10%
23400      FLAG% = 5% : GO TO 23760
23410      ! VALIDATE ADDRESS
23420      IF LEN(M$) < T10% + ADL%
           THEN FLAG% = 7% : GO TO 23760
23440      T10$ = MID(M$,T10%+1%,ADL%)
23450      AD% = FNOC$(T10$,ADL%)
23460      IF AD% < 0%
           THEN FLAG% = 7% : GO TO 23760
23480      !
23490      IF LEN(M$) < T10% + ADL1%
           THEN 23650 !INSTRUCTION COMPLETE
23510      T10$ = MID(M$,T10%+ADL1%,1%)
23520      IF T10$ = "X"
           THEN INDEX% = 1%
           ELSE IF T10$ = "I"
                THEN INDIR% = 1%
                ELSE FLAG% = 8% : GO TO 23760
23570      IF LEN(M$) < T10% + ADL2%
           THEN 23650 !INSTRUCTION COMPLETE
23590      T10$ = MID(M$,T10%+ADL2%,1%)
23600      IF T10$ = "I"

```


CIRCLE 54 ON READER CARD

A SECURITY CHECKLIST

By Dave Mallery

Physical: Start outside your computer room. Is the door locked? Is the key hanging there? If there is a window, what can you see from the corridor? Is the console right by the window? Can you read the print-out? Can you watch the operator log in? Can you see the disk drives? Are the packs labeled? Are the labels the MFD passwords?

Now take the 'VIP' tour of the computer room that the boss would give to visiting firemen. Repeat all of the above questions. Here are a few more: How many dial-in phone numbers can you find—look at the telephone blocks and modems. Look at the console closely—how many passwords can you find? Any written on the terminal? Are any of the terminals logged in—(try control 'T'). How quickly can you run \$REACT? (or \$ONLPAT).

After the cook's tour, it's time for some more methodical work. Make a list of every keyboard on your computer and the actual physical location. While doing this, it's a good time to actually look at each of them. How many are logged in and unattended. Are any in a privileged account? How many have passwords written or taped to them. Try the answerback escape sequence on every VT100 and equivalent and see how many contain log-ins. Now that you have a list and are satisfied with the contents, actually open the back of your machine, and physically verify that 'empty' ports have nothing plugged in. If you find a cable in an 'empty' port, follow it. It will lead to something interesting. Perform the same steps for all known telephone circuits and modems in your computer room. While looking at cables and connectors, make sure they only have one wire coming out and that there are no three-connector 'bridges' in sight. Now, take a full listing of all accounts on your system disk (\$MONEY). Sit down and identify the owner of each account. Scan your disks (all of them) for programs with protection codes conveying privilege. Identify each program and its reason for priv. Demand sources of all such programs and re-compile them periodically without notice. A very good check is to log into a non privileged account and look around via DIR at the privileged ones. Even if you have used the feature patch for restricting directory access.

Try this — Run \$RNO
RNO>KB:= \$ACCT.SYS

Try every program you can see that has a <232>. Go home some night and spend an hour trying to break in. How many CCL'S work logged-out. How much can you learn about your system before logging in? See if anyone challenges you. See if anyone notices. See if anyone cares.

Print out a list of all passwords. Besides DEMO, SYSLIB, SYSTEM, MFD, PACK01, and other obvious ringers (your name, the company name and initials...). How many passwords are two or three letters long? The best password is six long, contains at least one numeric digit and makes

very little sense. Most priv. accounts don't need passwords (exception: [1,2]) [1,0] & [1,3] through [1,5] especially don't if used as libraries. Get rid of them.

UT CHANGE [1,n]*

There are different schools of thought on the subject, but I believe only one privileged password is required unless substantial development work is going on in privileged accounts. (If you have one, you have them all...). One password is very simple to change. You should change it frequently. Once in a while, you should change it and wait to see who asks.

There are several other areas of security to consider. Systems that are physically secure may be vulnerable to psychological attack:

The phone rings at 9 PM. Your night operator answers. The caller identifies herself as Digital Software Support and asks if that emergency patch to the disk driver has been installed. Warning of imminent system crash, she has the operator install a patch over the phone that allows logged-out users to have privilege. (A one-word patch, by the way). Do you have a procedure for such probes. Does your personnel know how to validate such phone calls? Do you have a call-back procedure? How do you know that this patch has not already been put in place? Do you know how to find out? Do you have a procedure for departing employees?

Trojan horses are another consideration. A trojan horse is a hidden switch or program entry point in a piece of third-party software which could be used by an informed person to penetrate an otherwise secure system. There is really no defense here. It never hurts to require third-party vendors to certify that their software contains no such traps.

Finally, here's a list of things to remember when dealing with a system that has been penetrated.

If your system has been penetrated, consult both law enforcement authorities and DEC security, before you do anything.

All monitor SIL's, RUN-TIME SYSTEMS, LIBRARIES and all privileged programs must be re-generated and replaced. You must have a known secure distribution. Get a new one if there is the slightest possibility of compromise. Have the sysgen and patching done by an outside source.

Remember to isolate and save the suspect media. They may be useful as evidence.

In the end, the only really secure system is one that has been powered down. We would be very negligent in our duty to our employers if we neglected the obvious.



DECUS AUSTRALIA 1982

By Carl B. Marbach

Winter in July, day into night, the Southern Cross and water that goes down the drain the wrong way: Australia! Here we were in Sydney, New South Wales on our way to Melbourne, Victoria and the 1982 Australian DECUS meetings.

It was the middle of July, winter here, but this is the land of mild winters; it never snows in the cities and seldom goes below freezing. The days are a mixture of sun and clouds, with mild temperatures often reaching the upper 50's or 60's even this far south (it gets colder when you go south, away from the equator). Our daytime was night at home; a 14 hour time difference—not to mention a whole day since we crossed the International Date Line. My wife, Helen, has it all figured out: as you go farther west it keeps getting earlier and earlier until suddenly it is tomorrow! At least she understands.

So, after 21 hours in two different aircraft (5 hours to San Francisco via DC-8 and then 6 hours to Honolulu and 10 hours to Sydney in a Qantas 747) we were here. Hats off to Qantas, they run a fine airline; we ought to know after 16 hours straight with them. Friendly helpful people foreshadowing our experiences of the next two weeks in beautiful Australia.

The winter (!) sky contains some very unfamiliar constellations, and we wondered at all the ancients following the legendary Southern Cross. We can now confirm that high pressure rotates counter clockwise and water does go down the drain in a vortex the opposite way it does north of the equator.

Although Australia is about the same size as the U.S. most of its population live in coastal cities. Sydney is a major city on the southeastern coast of the continent. The downtown area is a mixture of old and new high rise office buildings, and the city is clustered into a neat center near the famous Sydney Harbour and Opera House. Culture abounds and the Opera



Up a tree Down under.

House lives up to its reputation with a complete opera hall, concert hall and drama theatre each specialized for the art to be presented. We toured the facility and had a lovely dinner overlooking the harbour. The scenic harbour is full of sailboats and other craft, including a hydrofoil commuter boat to other areas across a dramatic entrance to the sea.

Outside Sydney there are bedroom communities which give way to farms and rolling plains leading to the Blue Mountains, so named because of their blue color caused by the gas escaping from the eucalyptus trees lining the sides of the hills. The gorges, valleys and spectacular panoramas is an amazing sight. Sydney, which was now more than 60 miles away was still visible! I should not fail to mention the snowstorm that we rode into and out of going through one of the many high roads in the mountains, and in July too.

We promised to visit one of our hosts, Chris Brett of Uncle Ben's of Australia, on our way to Melbourne so we took an early flight to the twin cities, Albury/Wodonga. The twin engine Fokker was comfortable and unlike in the States, the captain was happy to have us visit the cockpit where he

could point out the scenery of the interior of the country. Landing in Albury/Wodonga is like homecoming; lots of families greeting returnees and the same saying goodbye, very rural and friendly. We were taken to the plant where Uncle Ben's manufactures pet food. The office space uses an open approach; everyone sits at his desk in a large open room (even the general manager) and when private conferences are needed you simply schedule one of the many such rooms located around the perimeter. We met with the programmers, managers and plant personnel for business and lunch which was followed by a plant tour. The day was crisp and clear, not easily forgotten, and a real look at Australia at work. After dinner in town we got the late flight to Melbourne.

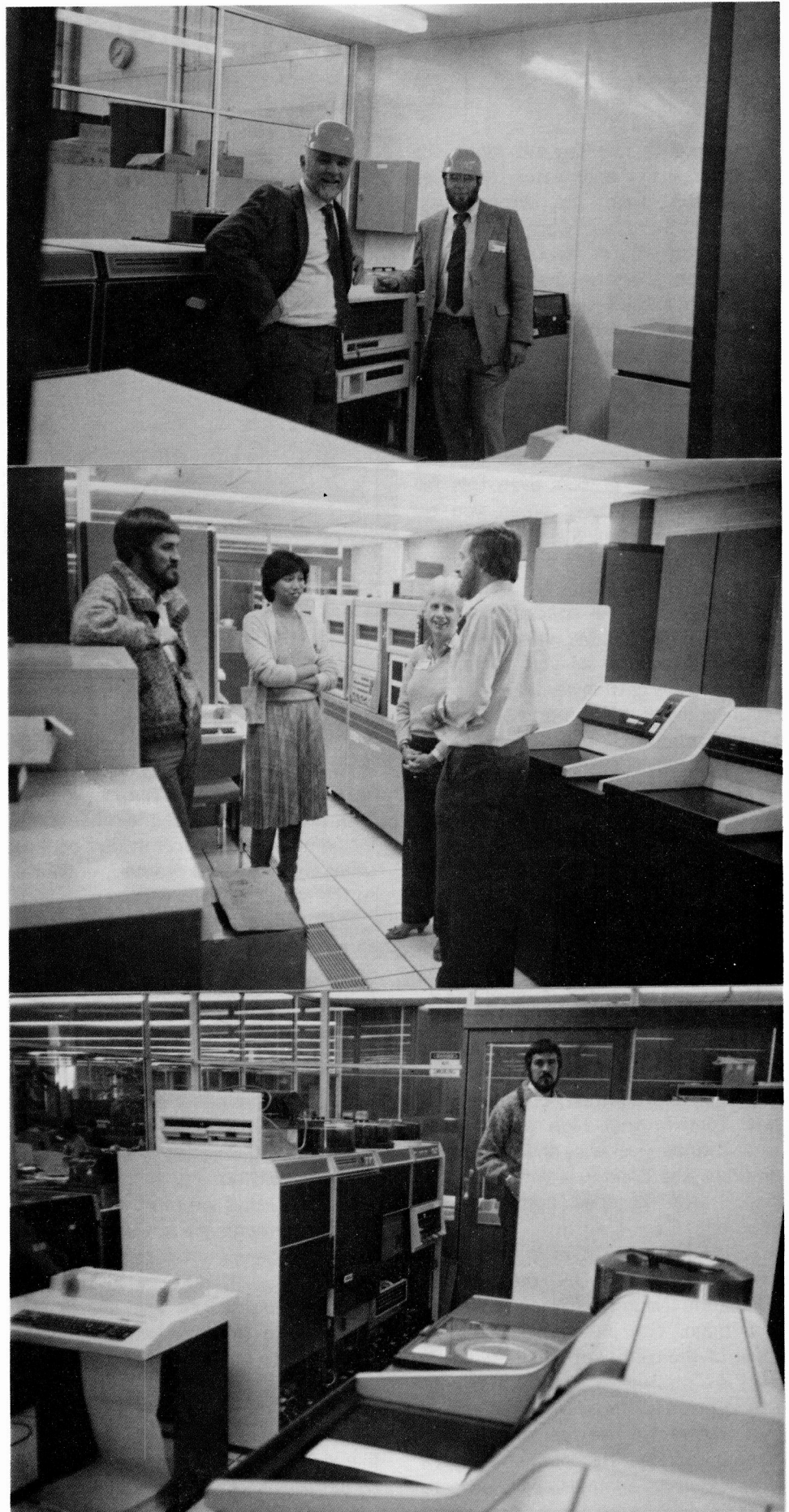
Melbourne appears as an older city, spread out with lots of minor cities (suburbs?) surrounding the center. It appears much larger in size than Sydney, but comparable in most other things. Which is the capitol? Neither, since they couldn't decide between the two cities, Canberra was chosen. Canberra is located almost exactly halfway between the two cities.

The Melbourne Hilton hosted this DECUS and is located beside two parks. One can ride the public transportation in clean, well lit friendly conditions. At rush hour they sell tickets at the bus stops so the driver can just pull up and load without long lines waiting to pay. Melbourne has a harbour and sights to rival any city in the U.S. To us it resembled a city on the move in the 1940's—all the good and none of the evil that the States' major cities have come upon in the 50's, 60's and 70's. There was no grafitti, no areas that were dangerous to walk around in and the people were very friendly and interested in us and our country. We even enjoyed watching Good Morning Australia when we woke up in the morning (evening at home). We make it a point to visit several special places

... to page 78

(Can you find the VT05?)

(Can you find the VT05?)

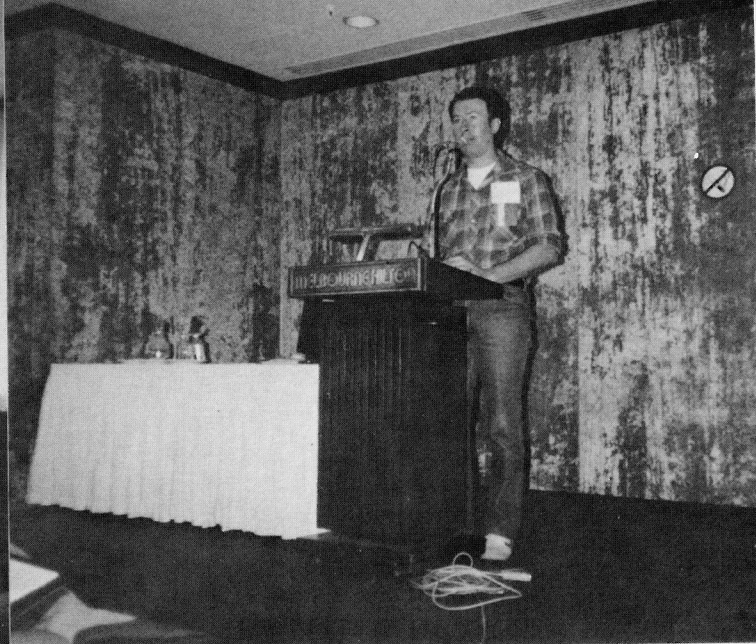




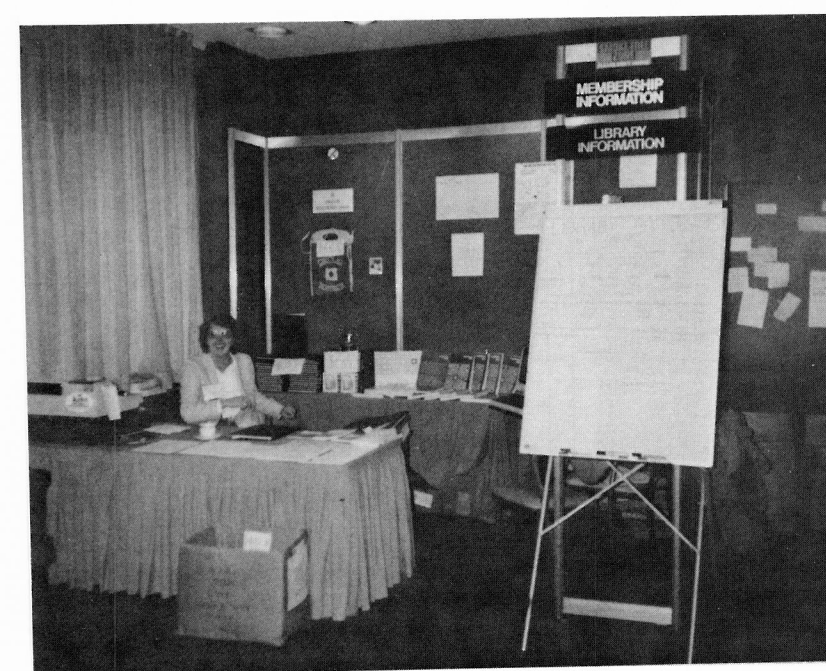
COMBOARD™ Software Results Corporation
DEC, UNIBUS™ Digital Equipment Corporation



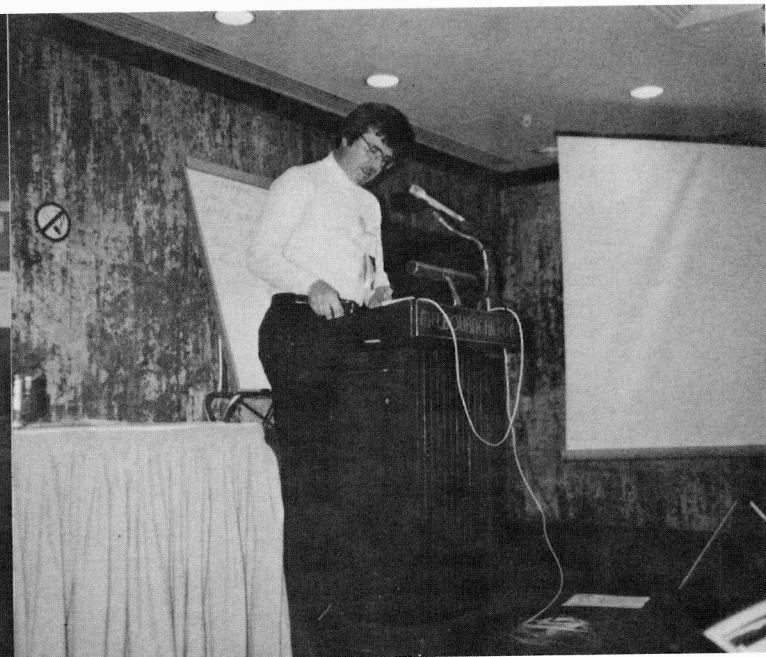
The exhibit area.



Chris Brett, next year's symposium RSTS Coordinator addresses the RSTS attendees.



DECUS Library Store.



Jo Kruithof, Newsletter editor, addresses the RSTS Group on "Non DEC Software Packages."

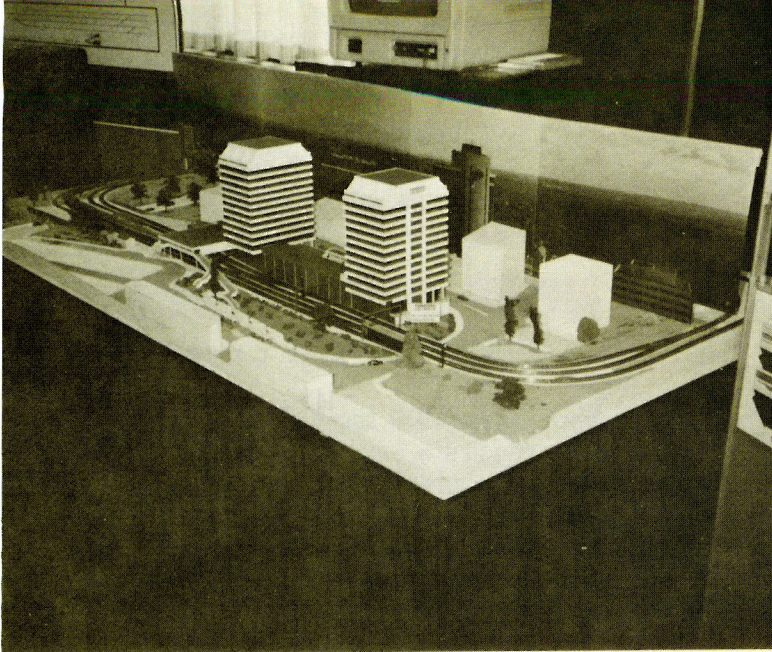
when we visit a city and one is the zoo! I particularly enjoyed the platypussary where we saw several swimming. The wombat was also a highlight as were the wallabees and koala bears.

DECUS symposia are much more relaxed outside the U.S. This may be because of the more realistic attitude adopted by most attendees that they are really THEIR meetings. Subjects are discussed that would not be allowed at the U.S. counterpart. While they also refrain from commercial activities, a session which I liked very much presented non-DEC software for RSTS by

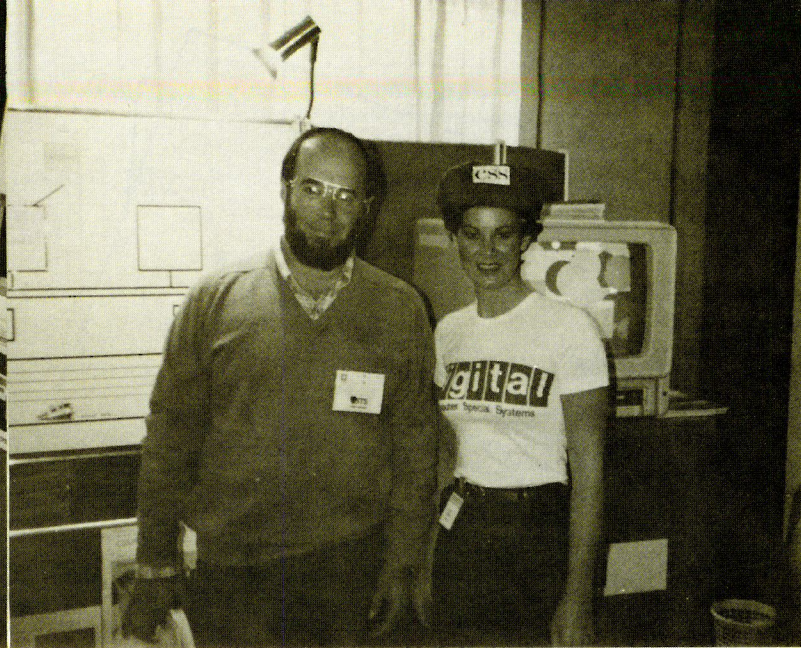
users of several packages available for RSTS. No prices were discussed, but we did manage to ask what it did, how well it did it, how the documentation was, how easy it was to use and what kind of support the vendor provided. In all my years (10) at DECUS this was among the most valuable; and I had to travel quite a distance to hear it! It was a real DECUS. The DECUS store sold souvenirs. DIGITAL had a large exhibit hall and there were meetings going on when I had to be at other meetings! The hallways and lunch tables were great places to meet people and talk DEC. I always learn so much from such

exchanges that it is a main reason for attending. For the RSTS group and others I suspect, Paul Konig from DEC fit right in and contributed much about DECNET, TECO and Version 7.2. The DECUS dinner was great fun, followed by some speeches, not all of which were very serious. I heard one story from a doctor in the North about. . . well, you'll just have to come to the next DECUS to find out. He does it every year.

After the meetings were successfully concluded we headed North (remember, it gets warmer when you



Computer controlled trains . . . Great for Winter in July.



Question: Who does she work for?



Yes, there will be PDP-11's for the 1980's.



The Australian Museum in Sydney, NSW.

go North) to Cairns on the northeast coast. We were able to visit an Aboriginal village and many hills and valleys in this lovely tropical area. Aside from relaxation, a major draw for this area is the Great Barrier Reef. This collection of reefs and small islands guards the coast of Australia for some 2000 miles and is alleged to be the "largest living thing in the world." The reef lies about 30 miles off the coast so you have to travel to get there. When the wind subsided enough, we were able to take a fast boat, called appropriately enough, Down Under, out to the reef for a day

of diving and snorkeling. While I found the visibility at 50 feet to be average (about 50-75 feet) for good diving waters, the abundance and variety of life was amazing. Everywhere you looked there was something alive; giant clams, coral, nudibranchs, fish of every size and shape and so much more that we must return and spend more than one day and two dives. While the snorkeling was nice, this is a place to be UNDER the water, a part of the reef itself.

It was finally time to go home, but not without a stopover in Brisbane, a fast growing "sunshine" city about

midway on the east coast. Like many sunbelt areas, Brisbane is undergoing rapid growth. The city hustles and bustles with excitement. We need more time to really experience the old and the new that it has to offer.

Too soon we were in the 747 cocoon that hums for about 10 hours and then disgorges you in a different land. It was a tiring but exciting two weeks in Australia. Thank you DECUS and the RSTS group for being such wonderful hosts. As usual we learned much, and we hope left some good ideas with you. So long for a while. We will be back. ♥

Western Maryland College
Westminster, MD 21157
Computer Center

Sir:
Can someone who knows more about it than I do please advise on the wisdom of installing RSTS V7.2? I have read the release notes and other information accompanying V7.2 and frankly didn't see anything significantly different from V7.1, or anything of importance to our system environment. This rather leaves me to wonder what the directions say the sysgen can't be done on-line, and our need to transfer 800 user accounts, not to mention patching—both DEC-inspired and in-house — makes the prospects of this exercise very unattractive. I do not want to install it unless there is a good reason why I should.

Sincerely,
Emily G. Johnston,
Director

Following closely on the heels of V7.1 is V7.2. Many people are asking why it took several YEARS to get from 7.0 to 7.1 and only a few MONTHS to get to V7.2. It appears that RSTS will move away from complicated and long patch kits, toward a system that will feature many more "maintenance" releases. VAX/VMS is released several times a year and this approach may be the way RSTS will go in the future.

With V7.2 there are some new things. Probably the most important, although not for everyone, is the full support of cluster libraries. Cluster libraries allow more than one resident library to share the same starting address; each library can be mapped only one at a time, but the "clustering" allows more program to be mapped within the addressing limit of the PDP-11. Suppose you wanted to map two 8K shared libraries: Without clustering you would need 16K of address space, 8K for each one and that doesn't leave much room for your program let alone RMS or something else. With clustering, each of the 8K libraries can share the same starting address, using only 8K worth of addressing space instead of 16K. By using multiple libraries it is possible to write larger programs than you could without "clustering".

Because of certain restrictions and limitations that each library must meet, not all libraries can be clustered. Currently, only COBOL-81 V1.1 and FMS-11 V1.5 are the only DEC libraries that can be clustered. Look for the third party software vendors to use this feature and maybe even provide the tools for us to "cluster" the non-clusterable libraries.

There are some DCL changes, but none that enhances the offering enough to make it really worthwhile as a stand-alone. DCL is good to start with because it will be upward

VERSION 7.2 OF RSTS/E NOW AVAILABLE

By Carl Marbach

compatible with other (VAX) systems. Older users don't need it as it doesn't add any functionality to RSTS. On VAX, DCL is a nice way to do lots of things but on RSTS it doesn't seem to be worth it.

There is support in V7.2 for the UDA50 controller and the RA80 disk drive. The UDA50 is the new intelligent (isn't everything?) controller for UNIBUS systems. For a complete discussion of these controllers and drives, see Volume 1 Number 1 of the DEC* PROFESSIONAL. For RSTS expect to see DU0: and DUX: as the new device designator for the UDA50 disks. When DEC will actually start shipping these controllers and the new RA80, RA81 and the RA60 is still a mystery, but it could be anytime.

More 22 bit addressing for the 11/23-PLUS is included in 7.2 allowing the 11/23 to have up to 4MB of memory. The problem with earlier 11/23 systems is that the DMA controllers who have direct access to memory did not support the 22 bit addressing; to do 22 bit addressing you will need the upgraded controllers and an 11/23-PLUS or see some other vendors for how to upgrade your present 11/23.

SYSGEN should be done off-line on a fresh disk. If you know what you are doing it is possible to do them on-line however. Note that some cusps like MAKSil, TKB and SILUS may change so make sure you use the new versions for your SYSGEN. We have successfully done this type of SYSGEN by copying the distribution medium to a separate disk drive and then after the dialogue phase of SYSGEN, we modify the SYSGEN.CTL file to point to the drive we have the SYSGEN stuff on instead of the SYSTEM disk. By modifying this control file we have complete control over the SYSBAT procedure and wind up with a new SIL on the extra drive that can be COPYied or moved in some fashion along with the new cusps to the system disk and then INstalled.

Send/receive has a few minor changes, with the biggest being the notice that SYS call 18 (small send/receive) will not be supported beyond V7.2.

DECNET V2.0 or DECNET phase III is supported. RMS V1.8 has not changed.

The monitor is now smarter about repositioning things when you add BUFFERS to the monitor. Before you had to manually reposition the memory allocation table.

There are some minor BASIC PLUS proposed changes that you should look at in the release notes even if you are not going to install V7.2. DEC wants you to move to 7.2 and has announced that support for V7.1 will last only 6 months.

So there it is, V7.2; should you install it? Some will and some won't, your decision should be based on needs relative to the new features, DEC support and standardization with the rest of the RSTS folks. We won't rush into it, but we'll install it soon.



LATE BREAKING NEWS

Infinity Software Corporation

2210 Wilshire Boulevard, Suite 801 Santa Monica, CA 90403 (213) 820-2702

Letters to the Editors
RSTS Professional

November 11, 1982

Dear RSTS Pro:

In the RSTS Professional of September 1981 Steven L. Edwards published three patches to the EDT version 2.0 release for RSTS/E version 7.0. Since that time there has been new releases of RSTS/E, and with them a slightly different release of EDT V2.0.

As published in volume 3.3 of the RSTS Pro, all the EDT patches will still work, with the exception of the patch to force VT100's back to VT52 mode on exit from EDT. The correct patch for the new release of EDT has changes to three lines.

The lines that read

```
IO.CXR = $$$LOC+742
IO.CFR = $$$LOC+766
. = $$$LOC+3360
```

should now read

```
IO.CXR = $$$LOC+756
IO.CFR = $$$LOC+1002
. = $$$LOC+3374
```

For those counting, the difference was increase in the size of the IOMOD module earlier on by 14(octal) bytes. The checksum for the new IOMOD module is 23007, the checksum for the corrected patch file is 21406.

For those desiring to install both the VT100 to VT52 patch from volume 3.3 of the RSTS Pro along with the journal name patch printed in this issue by David Spencer, make the following changes to the patch procedure.

Install the correct Steve Edwards patch, but do not replace it in the library nor task build. Enter the journal name patch file and modify the line that says

```
. = $$$$+003376
```

to become

```
. = $$$$+003466
```

Run PAT.TSK with the new checksums for IOMOD with the Steve Edwards patch being 16011, and the journal name correction file checksum being 22621. Remember not to select the same output file name for the IOMOD module as the input module, or PAT will fail. Replace the IOMOD module with the two applied patches in the library and task-build normally.

EDT devotees;

Steven L. Edwards
Software Techniques, Inc.

David Spencer
Infinity Software Corporation



COBOL
SOFTWARE SOLUTIONS
FOR PRODUCTION AND
FINANCIAL CONTROL

Available on
VAX
in native mode
PDP 11
under RSX or RSTS

OFF-THE-SHELF SUPPLY
Enquire about our
30-DAY TRIAL
ON YOUR MACHINE

IMPCON

FULL FACILITIES FOR:

General Ledger—
Accounts Receivable—
Accounts Payable— Sales Orders—
Stock Control—Purchasing—
BOM—Works Orders—MRP—
Work in Progress Monitoring

Send for
FEATURE WALL CHART
to:

**COMPUTER SYSTEMS
DEVELOPMENT (CSD) LTD.**

75 Marylebone High Street,
London W1, England.

Tel: 01-935 2255

NEWS RELEASES



PRODUCT UPDATE

EVANS GRIFFITHS & HART, INC. ANNOUNCED RELEASE OF VERSION 12 OF SELECT

Lexington, MA — Evans Griffiths & Hart, Inc. announced the release of version 12 of SELECT, its fast record extraction package for RSTS/E on the PDP-11. The package can also be used under VMS on the VAX with ROSS/V, EGH's RSTS/E simulation package.

SELECT is a machine-language package that employs user-specified selection criteria to extract from one or more input files those records needed for a specific report or application. Optionally, the extracted records are modified by the deletion, insertion, or rearrangement of fields. The records are then written to an output file that may be sorted by making use of SELECT's interface to FSORT3, EGH's fast machine-language sort package.

Version 12 of SELECT contains a new operator that allows the user to generate more than one output record for each input record. Multiple selection expressions control successively whether a given kind of record is output for each input record. This feature is most useful in situations, such as building multiple file indices, where making a single pass over a large input file may save considerable time over making multiple passes.

Another change in version 12 is that SELECT's front-end modules may be compiled under either BASIC-PLUS or BASIC-PLUS-2. Previously, the package's front-end modules could be compiled only under BASIC-PLUS.

A single-CPU license for SELECT is \$1500, discounted to \$500 when the package is purchased with FSORT3. OEM discounts are available. Contact Evans Griffiths & Hart, Inc., 55 Waltham Street, Lexington, MA 02173. Tel.: (617) 861-0670.

QUERY AND REPORTING TOOL INTERFACES WITH MANY PACKAGES

New York, NY — Enterprise Technology Corporation has released Version 2.0 of its Generalized Reporting System (GRS), a user-friendly query, report-writing and auditing package designed for use with a wide variety of applications software packages and database systems. GRS operates on Digital Equipment Corporation PDP-11 series computers under RSTS/E. A version of GRS for VAX series computers is scheduled for release in December 1982.

The new version of GRS is compatible with the applications software packages distributed by NCA Corporation, Martin Marietta Data Systems, Transcomm Data

Systems, Interactive Management Systems and AMCOR Computer Corporation, among others. GRS has also been interfaced to the TOTAL and AMBASE database management systems, and to the RMS-11, DMS-500, DIBOL, FMS-11, and FAM file management systems which are used under RSTS/E.

GRS consists of a powerful English-like query language which is suited for use by both DP and non-DP users. The language is designed to be used by management personnel to meet their own needs for ad-hoc information retrieval, and is also intended to be used by systems personnel to vastly decrease the amount of time spent responding to user requests for reports. GRS also includes many features which make it ideal for use by auditors.

The new version contains a CHAIN interface, allowing GRS to be called from other programs and from user menus. Other features in the new release include support for the PDP-11 floating-point processor, support for larger data records than previous versions, an option to utilize record keys to speed access to selected data, and enhanced facilities for the use of "canned" command files. It also includes expanded user documentation.

The GRS query language is designed so that one statement is all that is necessary in many cases to produce a nicely-formatted report, including page numbers and column headings. This is accomplished through the use of a wide range of defaults, all of which can be overridden by the user if desired. GRS provides the flexibility to generate reports in practically any format desired. Summarization statements and an advanced set of built-in calculations allow the user to aggregate data and perform complex analyses, again using a single statement in many cases.

The GRS user can easily supply his own titles, headings and page footers, and several different files, record formats and/or print line formats can be incorporated into a single report. GRS includes an extensive online user Help facility and optional password protection at the individual field level. Users can save sets of commonly-used GRS commands in command files, thus allowing even the most complex report to be produced by typing a single word to GRS. The system includes an easy-to-use editor to aid in preparing and changing these command files.

Additional GRS system features include a high-speed sort facility, multiple control breaks with totaling and subtotaling, weighted averages and other summary calculations, table look-ups, elementary statistical analysis, and random sampling. The user can enter specialized calculations in standard algebraic format, and the results of these calculations can be used to create new, "temporary" database fields or to temporarily redefine existing fields. This redefinition capability is useful in constructing forecasting applications.

GRS contains a facility which allows ETC or a customer's systems staff to tailor GRS to meet the particular requirements

of an installation. For example, GRS can call a variety of database access routines in the preparation of a single report, including access methods which the customer has supplied and incorporated into GRS. Programmers can also code their own conversion modules for inclusion into the system, in order to handle specifically encoded or encrypted fields. To facilitate its use with complex databases, GRS is capable of working with several different record formats at once, determining any given record's format by examining the record itself.

It is not necessary for the user to describe to GRS the fields which comprise the record(s) he is working with; the database description process is performed only once, when GRS is first installed at a customer site. During a GRS session, the user needs only indicate the names of the data fields which are needed for the report he is preparing.

GRS operates on most types of interactive terminals, both CRT and hard-copy. A special Scope mode allows GRS to pause at the bottom of each page, thus facilitating its use on video terminals. Users can optionally direct their output to alternate hard-copy devices or to listing files for later printing.

The GRS software is supplied under a perpetual license for a one-time fee of \$8,500.00 which includes full documentation and one year of support. Extensive training and customization services are also available. More information can be obtained by contacting Enterprise Technology Corporation at 305 Madison Avenue, New York, NY 10165, (212) 972-1860.

COMPANION PRODUCTS FOR DEC COMPATIBLE CT*OS WORD PROCESSING SYSTEM

Pasadena, CA — Office Automation software house, Compu-Tome, Inc., is now offering two additional software products as companions to their DEC compatible CT*OS word processing package.

The first product, CT*OS spelling corrector, works in conjunction with CT*OS word processing and provides very high speed, 80 words per second, checking of spelling in documents several thousand words in length. The corrector, working with a dictionary of over 80K words, actually allows the user to correct misspelled words, rather than merely flagging them as in many less powerful systems. Both the speed and file efficiency are the result of a unique design based on a compressed syntactic format. If the word is not encountered in the dictionary, it is stripped of common prefixes and suffixes and checked for root comparison.

The second product is a powerful management communication system dubbed Exectronic™ Mail. Exectronic Mail will support up to 64K users on a single network and operates under DECNET protocols. Besides the huge number of user notes, Exectronic Mail is unusual in that it is management oriented, designed to facilitate communications among decision making users. Thus, an active response is required of both sender and receiver; i.e.,

The structure of the system is based on a "routing slip" format which makes it "friendly" to users familiar with the ubiquitous office routing slips.

The system, operating with CTOS as the resident editor, offers generous file sizes and fonts; for example—up to 1K messages per user stored on-line, names and titles up to 53 characters. Storage demand is minimal; 24Kb for the first user, plus 6Kb for each additional user up to system limits.

The corrector and Exeonic Mail operate under RSTS/E, RSX-11M, RSX-11M+ and VAX/VMS and are fully integrable with CT*OS. An important application is the ability to run with DEC's new "300" series personal computers acting as an intelligent node on a network.

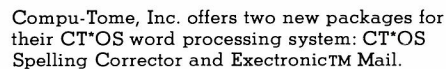
Compu-Tome, Inc., 234 E. Colorado
Blvd., Pasadena, CA 91101, (213) 796-
9371.

GEJAC ANNOUNCES ARSAP FOR RSTS
Riverdale, MD — The ARSAP Resource Management and Chargeback System, has been updated to run on PDP-11 computers using RSTS versions 7.0, 7.1 and DEC's recently released 7.2 operating system.

ARSAP is now available on all the most widely used DEC operating systems: RSTS, RSX-11M-PLUS and VAX/VMS. Managers of multiple operating system shops can get compatible reporting of resource utilization, making it easy to compare productivity on all in-house DEC systems. This compatibility has not been available before now.

1. **Project Accounting** — Users can be prompted for a project-id and password upon logon and the terminal session can be allocated to his project as well as his logon account, so it is easy to monitor projects and users.

terminal, showing the activity on each terminal, when the terminal is used and the amount and type of resources consumed. The reports are presented in graphic as well as numeric format.



Compu-Tome, Inc. offers two new packages for their CT*OS word processing system: CT*OS Spelling Corrector and ExelectronicTM Mail.

4. **Invoicing** — Invoices can be generated by user or by project. Rates can be by time of day and can be set by individual user, project, or system default basis.

ARSAP also can produce itemized invoices automatically for allocating costs of computer resources to internal departments, projects, users, contracts and grants. It is used for departmental budgeting, to control project costs, or to bill timesharing customers. ARSAP's equitable billing system enables bidding competitively on contracts and satisfies government and commercial job costing requirements.

Contact: GEJAC, Inc., P.O. Box 188,
Riverdale, MD 20737, 301-864-3706.

London, ONTARIO — Cablesare Inc. has announced PORTAL - VMS, designed to alleviate the need to transfer files using

PORTAL-VMS may also be used in conjunction with Cablesare's LSI-X.25 interfaces to provide file transfers over private and public packet switched networks.

For more information, call or write:
Barry G. Walker, Director of Marketing,
P.O. Box 5880, 20 Enterprise Drive,
London, Ontario N6A 4L6, (519) 686-
2900.

Los Alamitos, CA — At a meeting of the Orange County Water Association today, Software Techniques Inc. announced a new integrated Billing and Accounts Receivable system designed to meet the special needs of water companies. Named WBS-11, the new system is designed for DEC PDP-11 computers running the RSTS/E operating system.

According to Karl Kemp, Manager for Mesa Consolidated Water District where

the new system is installed, "It is a custom-made system. The entire program was designed specifically to meet the particular needs of a water agency. I am excited about this new addition and look forward to what it can help the District accomplish."

The system offers easy access to all functioning through a state-of-the-art menu selection, which also provides extensive security options. "User-friendly" features include the system's ability to accept data in a variety of ways and arrange it into a standard format. In addition the system can access account history by soundex, providing customer information on anyone with a name or street even sounding like the one in question.

WBS-11 is available directly from Software Techniques, Inc. (Software Techniques, Ltd. in the U.K.) and from authorized distributors.

VAX USERS GET SESSION DOCUMENTATION

Provo, UT — Clyde Digital Systems, Inc., a principle supplier of system utilities, announces the release of CATCHIT, the most unique documentation tool available to VAX users. For the first time, VAX users now can achieve accurate and reliable run examples while executing interactive tasks on the system.

Eight control characters make up the easy-to-use yet powerful command set. By running CATCHIT, all terminal input and computer output is captured in the log file. All user input is underlined as well. CATCHIT does not lose terminal prompts, and catches all terminal I/O that comes through.

Use CATCHIT for audit trails, user documentation, or detecting errors when documenting programs. At any time you may insert descriptive text into the log file. This text has no effect on the current session, but is simply placed in the log file. This is particularly valuable for including explanations of what you are doing in the resulting document so that others will be able to read it and do what you did.

Users familiar with DOC, the powerful RSTS counterpart, will immediately recognize the advantage of enhancing their computer system with CATCHIT. The user community will quickly recognize its versatility and attractive pricing. CATCHIT may be licensed for a one time fee of \$600.

For more information, call or write LeAnn at: Clyde Digital Systems, 3707 North Canyon Rd./3-E, Provo, UT 84604, (801) 224-5306.

CURRENT SOFTWARE ITEMS AVAILABLE

Columbus, OH

- 1) CCLMAN - CCL Manager — Enhanced from the June, 1982 RSTS PROFESSIONAL - \$15.00.
- 2) ..CB.. - CB Communications System — Greatly enhanced from the August, 1982 RSTS PROFESSIONAL - \$15.00.
- 3) RTS - How To Write a Runtime System — with Example Runtime System - \$15.00.
- 4) CCLMGR - Runtime System Interface to

CCLMAN - \$15.00.

5) USRDSK - System Manager user disk sortage report - \$15.00.

Send tape and tape format required ...

If you do not enclose a tape, please enclose an extra \$10.00 to cover the cost of the tape.

For more information contact: Philip Hunt, O.L.F.B.P., 6400 E. Broad St., Columbus, OH 43213, (614) 864-9200.

MDB INTRO'S BUS REPEATER FOR LSI-11 USERS

Orange, CA — A Bus Repeater that allows LSI-11 Q-Bus processors to be extended beyond their 20 DC bus load limitation, has been introduced by MDB Systems, Inc., the world's largest independent manufacturer of minicomputer interfaces.

Called the MLSI-DB11-R, it receives and redrives all of the computer bus signals so that any type of devices can be on the expanded bus which supports up to 19 additional DC loads. Developed in response to the trend toward larger systems capability, this new generation Bus Repeater supports 22-bit addressing, 4-level interrupts, Direct Memory Access modules and parity.

Design features include terminators and two dual height modules, one for each bus segment. They plug directly into LSI-11 processor backplane and expansion chassis and provide the required 120 ohm termination for the CPU bus and matching impedance for the beginning of the second backplane.

The MLSI-DB11-R is software transparent and is fully bi-directional as signal direction may change several times for each bus cycle. Program array logic in the MLSI-DB11-R monitors bus activity and issues gating commands to handle address, data, interrupt and DMA control signals. As a result, the CPU is unhindered by any device placed between it and other peripherals or memory in the system. An edge mounted LED indicates when the repeater is active.

The overall design of MDB's MLSI-DB11-R allows large, single CPU systems to be implemented with one or more bus repeaters, according to MDB marketing manager Stan Margulis, who adds that MDB's new Bus Repeater also provides an excellent method for future system expansion.

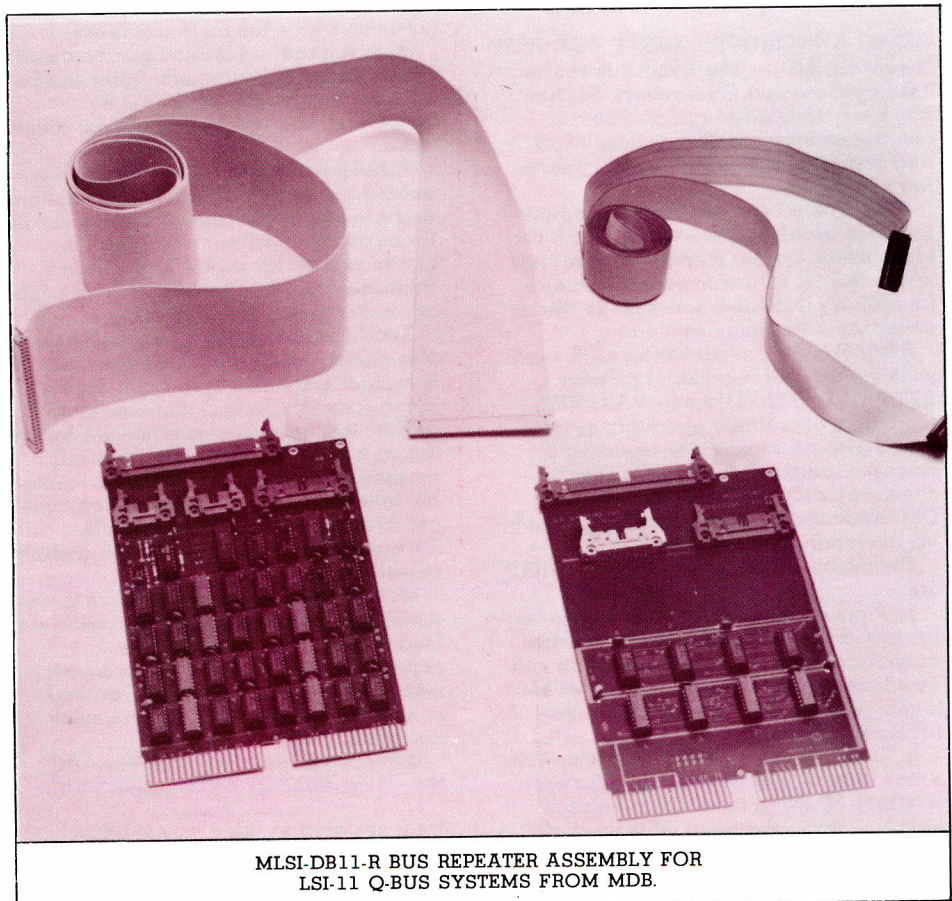
The MLSI-DB11-R is priced at \$900. in single quantity and delivery is available within 30 ARO. The price includes a 10 foot (3.05m) interconnection cable set.

NEW SOFTWARE

SYSTEM PERFORMANCE HOUSE ANNOUNCES SOLUTION TO DISK BOUND RSTS/E SYSTEMS

Columbus, OH — An easy-to-use solution to disk bound RSTS/E systems has been developed by the System Performance House, Inc. The program, known as DOPTEP, produces an optimized disk structure that increases system performance up to 50 percent.

"Our users have found that DOPTEP is superior to any other disk management



MLSI-DB11-R BUS REPEATER ASSEMBLY FOR LSI-11 Q-BUS SYSTEMS FROM MDB.

Further information about DOPTER can be obtained by writing the System Performance House, Inc. at 5522 Loch More Court, Dublin, Ohio 43017, or by calling (614) 265-7788. The System Performance House specializes in the development of technically superior utility software for DEC computers.

According to Manus, buying hardware is not always the way to solve a performance problem. Buying hardware to solve a tuning problem will only result in a tuning problem on a more expensive computer system. RESPOND's performance measurement tells the computer manager where to best apply his resources. It tells whether more hardware is needed, and what kind of hardware will

Manus Services Corp., Lake Union Bldg.,
1700 Westlake Avenue North,
Seattle, WA 98109, 206/285-3260.

At the Exposition, which was also open to DEC users who are not IRUS members, attendees could learn about many of the products and services that are compatible with their systems. Exhibitors included Able Computer, Accudata of Maine, Advanced Computer Systems, Amcor Computer Corp., Aviv Corp., Ball Systems, Braegen Minicomputer Peripherals, Cambridge Digital Systems, Data Processing Design, DTI Data Terminals, EDP Supply, Emulex Corp., Evans, Griffiths & Hart, General Electric Co., Hardcopy

IRUS serves its members with a monthly newsletter, product surveys, and holds regular regional meetings in New England, Southern California and Metropolitan New York. The next national conference is scheduled for the Fall of 1983. For further information contact IRUS, 3657 Post Road, Suite 4, Warwick, RI 02886 (401) 738-4430.

Of those who already owned their hardware but were seeking software for new applications, the greatest number, 41%, were owners of IBM equipment. The

Send news releases to: RSTS News Releases,
P.O. Box 361, Ft. Washington, PA 19034-0361.

CLASSIFIED

\$1⁰⁰ per word, first 12 words free with one year's subscription. [Be sure to include a phone number or address in your message.]

DEC BEST VALUES

PRE-OWNED DEC EQUIPMENT

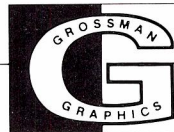
BUYING AND SELLING

SYSTEMS • CPU's • PERIPHERALS • TERMINALS
OPTIONS • MEMORY • COMPATIBLES

CALL DICK BAKER (305) 979-2844

dataware
incorporated

Carico Center
2845 NW 62nd Street
Ft. Lauderdale, Florida 33309
Telephone (305) 979-2844



5041
frankford
phila., pa.
19124

215
537-0782



**SUPER
CHARGED
SOFTWARE**

Increase your PDP 11's Performance 30-50%!

Supercharged Application Programs are Command Data's enhanced version of proven software which turns out more work, responds faster and extends the life of your PDP 11 investment.

Systems for: General Businesses, Contractors and Distributors.

Fully integrated and ready for use

AP • AR • Equipment and Asset Accounting • GL • Job Cost •
PR and Order Processing.



For details send us this ad with your business card:

Command Data

AUTHORIZED
digital
COMPUTER DISTRIBUTOR

165 West Valley Avenue Birmingham, AL 35209 205-942-3156

RSTS WRITER
B2 RMS \$99 mo.

ACT 4

603-347 BAY STREET
TORONTO ONT. M5H 2R7

BACK ISSUE OFFER
ALL 14 BACK ISSUES OF
THE *RSTS PROFESSIONAL*
\$95.00

Send check to: RSTS PROFESSIONAL,
Box 361, Ft. Washington, Pa. 19034-0361
— Payment Must Accompany Order —

RSTS CHEAP!

\$350/PORT, \$20/MB

on **WALL STREET. NYC**

Local error-free dial-in
or leased line termination

and in **PHILADELPHIA**

DEC 11/70, 24 hrs, 7 days

Secure Backups

B+2, DIBOL, COBOL81

DATATRIEVE, WORD-11, EDT

Secured by LOCK-11

NATIONWIDE DATA DIALOG

NATIONWIDE DATA DIALOG
DAVE MALLERY 215-364-2800



Dataram goes to extremes to satisfy LSI-11 users.

	DATARAM B23 PLUS	PDP-11/23 PLUS	DATARAM M23	PDP-11/24	DATARAM W23
CHASSIS	5¼" Front loading	5¼" Front loading	5¼" Rear loading	5¼" Side loading or 10½" Top loading	10½" Top loading with 80 MB Winchester
BACKPLANE	8 x 4	9 x 4	9 x 6	9 x 6	9 x 6
I/O MAPPING	None	None	Q-MAP	KT24	Q-MAP
BUS	22-bit Q-BUS	22-bit Q-BUS	22-bit Q-BUS and 18-bit Q-BUS	22-bit extended UNIBUS (EUB) & 18-bit UNIBUS	22-bit Q-BUS and 18-bit Q-BUS
MEMORY & PERIPHERAL SUPPORT	Up to 4.0MB, but limited to 256KB when using peripherals other than RL01/RL02 or RX01		Up to 4.0MB. No basic limitation on peripherals.		
PRICING	B23, M23, and W23 are approximately 35% less than comparable DEC systems in 1.0MB configurations. Additional 1.0MB memory modules from Dataram are about 40% less than DEC's equivalent 1.0MB memory. Contact us for actual price comparisons.				

DEC, LSI-11, PDP and UNIBUS are registered trademarks of Digital Equipment Corporation. Q-MAP is a trademark of Dataram Corporation.

From Q-BUS Pricing to UNIBUS Performance



From top to bottom, chart the range of LSI-11 system performance you get from Dataram...and only Dataram.

It starts with our low-end B23 PLUS Q-BUS system at lower (much lower!) than DEC prices. And continues with high-end M23 and W23 Q-BUS systems that use Dataram's innovative Q-MAP™ I/O mapping module to generate a separate 18-bit bus from the LSI-11/23's 22-bit bus. Allowing you to put 4.0MB of memory on the 22-bit bus while interfacing your high-performance peripherals (RM02, TM11, RX02 and more) to the 18-bit bus. Giving you much more performance than provided by the PDP-11/23 PLUS, which supports only the RL01/RL02 and RX01 on its 22-bit bus.

It's possible, because Dataram's Q-MAP duplicates the functions of DEC's KT24 — which exists only in UNIBUS minicomputers from DEC — enabling Dataram's 4.0MB M23 and W23 minicomputers to be completely compatible with RSX11-M, RSX11-M PLUS, RSTS, UNIX and any other operating system which supports KT24 memory management.

Our W23 system goes a step further, integrating an 80MB Winchester drive and associated controller to provide even greater performance.

Q-BUS pricing...UNIBUS performance...in an LSI-11 megabyte system. Plus a wide range of disk and tape controllers, and related LSI-11 accessories. Only from Dataram. Call us at (609) 799-0071. We'd like to tell you more about our family of LSI-11/23 based systems.

**DATARAM
CORPORATION**

Princeton Road
Cranbury, New Jersey 08512
(609) 799-0071 Telex: 510-685-2542

CIRCLE 55 ON READER CARD

EMULEX TALKS DEC

1

THE DISK WHIRLS...

Can your VAX 750's or 780's benefit from the high transfer rates of DEC's UDA-50? Emulex thinks not. Do your 750's or 780's need another Unibus adapter when the UDA-50 is used? Emulex thinks so. These, and other UDA-50 questions, are discussed in a special Emulex report. Write to us for your free copy.

MAINLINE TO MEMORY...

Did you know you can mainline directly between VAX-11/750 memory and almost any of the new high performance disks? The Emulex SC750 controller board plugs right into the CMI bus—bypassing the overworked Unibus. Or, if you want to enhance the performance of your VAX-11/780, plug an Emulex V-Master/780 into the SBI bus and mix and match up to eight high performance disks with individual capacities of up to 675 megabytes.

Note: The SC750 and V-Master/780 are the only controllers available today that'll handle the Fujitsu Eagle's 1.8 megabyte transfer rate.

BREAKING THE BARRIERS...

Frustrated over the cost and complexity of adding on-line terminals to your LSI-11? Is the multi-user potential of the LSI-11/23-plus just a tantalizing promise—thwarted by cycle-stealing, interrupt-driven DLV11's and DZV11's? Break the barriers with Emulex's new CS01/H communications multiplexer, a Q-bus multi-line interface with DH11 performance capabilities. Up to 64 (that's right, 64) channels attached to a single quad board, plus high-speed transfers. And features that even DEC's DH11 can't equal, like internal self-tests and transmission speeds up to 19,200 baud.

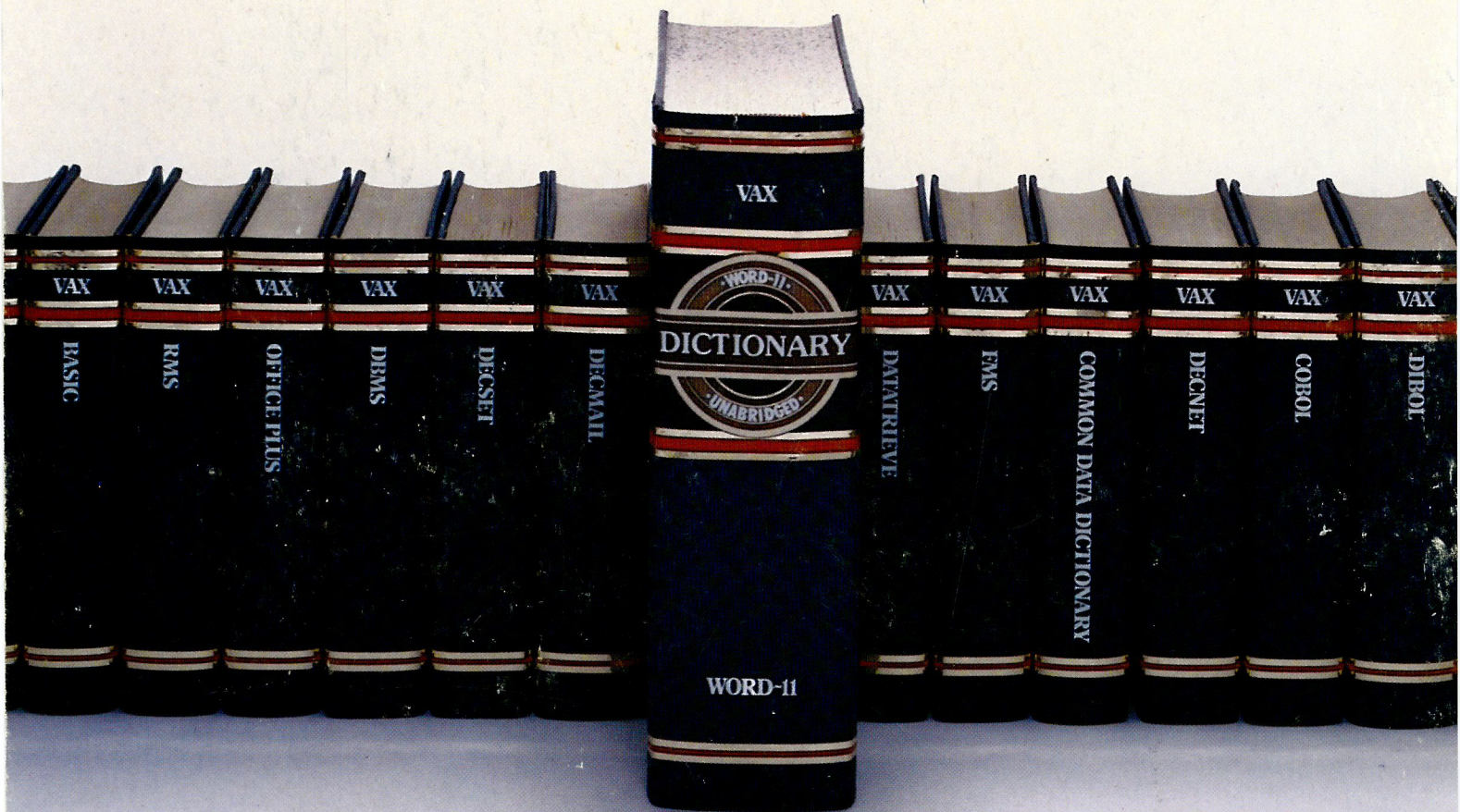
EMULEX SPELLS RELIEF...

Attention LSI-11, PDP-11, and VAX-11 users who feel bottled up by their present data storage capabilities. You can take advantage of those powerful new disk systems you've been coveting. Over 38 ready-to-ship disk subsystems you can simply plug in and run.

FROM THE EMULEX FILE...

It's been a 60 percent growth year for Emulex; details are in our annual report, available from the Company. New domestic and foreign sales offices. One hundred percent increase in foreign shipments. Construction is nearing completion on a new 71,000-square-foot headquarters building in Costa Mesa, California. Plus, Emulex's 11,500-square-foot plant in Dorado, Puerto Rico, is on line. Meanwhile, write to us at 2001 East Deere Avenue, Santa Ana, California 92705. Or, better yet, let's talk DEC—toll free. Dial (800) 854-7112 or (in California) (714) 557-7580.

WORD-11. The Word Processing System Vital For VAX.



The Unabridged Addition.

No matter how many additions you've made to your VAX,TM it won't be complete without WORD-11.TM It's the sophisticated word processing system designed to help you use all the other information you've stored. With features like list processing, built-in dictionaries for spelling error detection, automatic table of contents and footnoting that make report writing a snap.

WORD-11 runs concurrently with data processing on multiple terminals. It's easy to use. And it's been up and running in hundreds of installations all over the world for years.

WORD-11 is easy to install and operate. It's

cost effective. And it's supported by a dedicated team of experienced specialists. It could be the most important addition you make to your VAX installation. For details, Call, (714) 993-4160.

WORD-11 is also available for Digital's RSX-11M,TM RSX-11M PLUSTM and RSTS/ETM operating systems.



Data Processing Design, Inc.

AUTHORIZED digital COMPUTER DISTRIBUTOR

CORPORATE OFFICE
181 W. Orangethorpe, Suite F
Placentia, CA 92670
714-993-4160 Telex 182-278

NY OFFICE
420 Lexington Avenue, Suite 633
New York, NY 10170
212-687-0104

WASHINGTON D C OFFICE
4520 East-West Highway, Suite 550
Bethesda, MD 20814
301-657-4098

WORD-11 is a trademark of Data Processing Design, Inc.
VAX, RSX-11M, RSX-11M PLUS, and RSTS/E are trademarks, and remaining titles are products or trademarks of Digital Equipment Corporation.