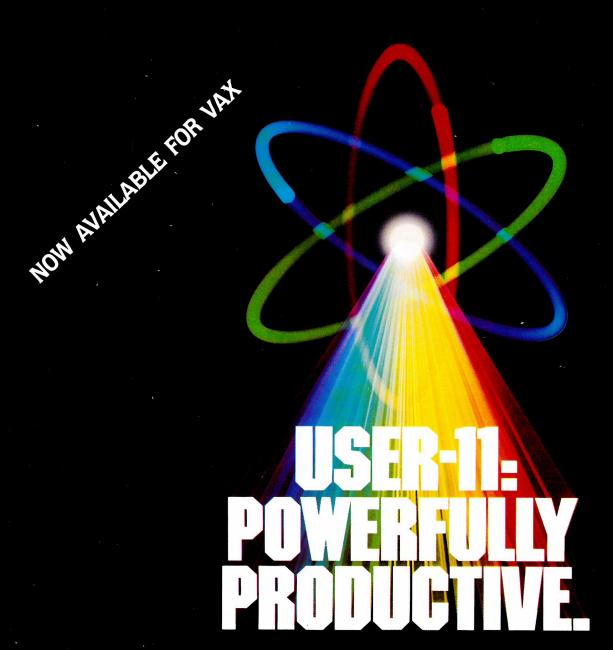
VAX PROFESSIONAL RSTS

Volume 5, Number 6

Anniversary Issue

December 1983 \$1000/issue, \$3500/year





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 simply describe and assemble these modules to create custom business packages—at an unprecedented rate.

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 (619) 745-6006, Telex: 182773

*USER-11 is currently available for DEC computers using the RSTS and VMS operating systems.

©NCCS 1983



Why settle for half the VAX information management solution?

Now there's a complete set of data and text management tools for VAX™ users. It's called SMARTSTAR™ and it lets you design completely integrated applications—from the physical VAX RMS file level to the user's application window.

This is the *first* software package for the VAX that lets you create multiple window applications, including a unique scrolled forms concept to greatly increase the effective VT-100[™] screen size. Components include:

SMARTDESIGN™ —
Design simple to complex applications, with no procedural or non-procedural code required.

SMARTQUERY™—Run those applications with linked windows, general query capability (including pattern matching), and with multi-user concurrent processing. All this still without code.

SMARTCALL™—Lets
you integrate SMARTSTAR's
fourth generation capabilities
with complex applications when
VAX host programming is
required.

SMARTSTAR is built around VAX architecture, so everything is consistent with Digital's standards. Interfaces to Digital's Datatrieve™ and ALL-IN-ONE™ are provided with general VAX datatype support.

CIRCLE 213 ON READER CARD

Get the complete solution. Call now for more facts about SMARTSTAR or any of our other VAX software products.



Signal Technology, Inc.

The leading independent software developer for VAX/VMS.™

5951 Encina Rd. Goleta, CA 93117 Call toll-free (800) 235-5787 California (805) 683-3771

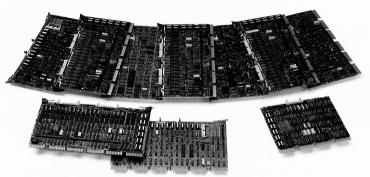
Germany • TCAE GmbH, Tel. 8139/6067, Telex. 527523 Sweden, Norway, Denmark, Finland • 3K Tre Konsulter AB, Tel. 0764/30175, Telex. 15559

Switzerland • Computer-Graphics AG, Tel. 1/932 34 82, Telex. 875447

Japan • Rikei Corp., Tel: 3-345-1411, Telex: J23772 U.K. • Logica Ltd., Tel: 09326/7355, Telex: 267413

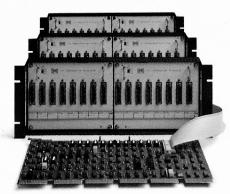
VAX, VAX/VMS, VT-100, Datatrieve and ALL-IN-ONE are trademarks of Digital Equipment Corporation.

Five reasons why DEC users should buy Emulex communications controllers.



Broad product line featuring our new DMF-32 emulation.

Nobody covers LSI-11, PDP-11, and VAX-11 users' needs like Emulex. More than 15 software-transparent controllers emulating DH11, DZ11, DV11 and DMF-32. All deliver improved line-handling capabilities, in a smaller package, at lower costs.



More channels.

Emulex's new DMF-32 emulation is typical. One controller board handles up to 64 lines, vs. only eight per DEC module. And Emulex offers *all* lines with modem control, not just two. For even more lines, Emulex's Statcon Series is the answer. We simply add a low-cost port concentrator, so that with one controller board you can connect up to 256 remote *and* local terminals.



Easy growth path.

As your system grows, upgrading is simple with Emulex controllers. Just change PROM sets. Example: DH to DMF for \$350. In addition, Emulex's advanced microprocessor architecture is consistent throughout the product line. Think of the inventory savings.



Fewer backplane slots.

Emulex communications controllers pack so much capability onto each board that fewer boards are needed. Take a 64-line DH11 emulation. Emulex does on one board what it takes DEC to do on 36. Think of the savings in rack space, to say nothing of price.

Lower prices.

For instance, a DEC DH11 controller lists at \$8,950 per 16 lines, with expansion chassis costing \$3,000 or more. Compare that to Emulex's CS11/H at \$4,500 for the first 16 lines and \$3,000 for each additional 16 lines. At 64 lines, you suddenly have savings of about \$23,000 and a lot of extra slots to boot.

Don't speculate with your communications controller dollars. Invest in Emulex. Phone toll free: (800) 854-7112. In California: (714) 662-5600. Or write: Emulex Corporation, 3545 Harbor Blvd., P.O. Box 6725, Costa Mesa, CA 92626.



The genuine alternative.

GSA Contract #: GSOOK8401S5575

Contents

VAX/RSTS SECURITY REVISITED Dave Mallery Security is not just packages or system features; it is a state of mind. These are so of the things you should be thinking about.	
A FEW NOTES FOR INFORMATION ARCHITECTS Al Cini First there were software engineers, now we have "information architects." Where have all the programmers gone?	. 14
PDP11/70 RSX11M-PLUS TO VAX 11/780 MIGRATION Allen Watson Migrating from RSX to VAX should be easy.	. 2
MicroVAX 1 and VAXEIan: A Realtime Marriage	. 32
THE PRICE OF SLOTH/EDT Ilan Caron Can an aborted EDT session be recovered? It is possible to save two hours of EDTing? Yes? No? But	. 36
THE VAX DRIVES THE COUNTRY'S 1ST FULL CHANNEL TELETEXT SERVICE Marci Shatzman You know what a VAX is, but do you know what Teletext is?	. 38
RESOURCE ACCOUNTING UNDER VAX/VMS — Part 1	. 43
AN ALTERNATIVE TO DATATRIEVE Ann Sangpan Allowing users access to their own data can be difficult. Can they really generate their own reports with or without Datatrieve?	. 50
INTRODUCTION TO THE VAX — A First Time Users Guide Terry C. Shannon New users of any computer system need all the help they can get. All too soon we forget what it was like when we couldn't even "log-in." Taking a step back to help the novices is a real talent.	. 54
KIM. Program to Transfer Files To and From TU58 Cartridge Tapes	. 68
COM.BAS Alton O. Moore, III COMmunicate with another computer system? Good idea.	. 77
THE ALL SINGING ALL DANCING SKELETON PROGRAM John D. Papadopoullos Extract all monitor information with a single program? Yes sir!	. 80
VAX SYSTAT: Doing It The RSTS Way Bob Stanley Can VAX really look like RSTS? Would you want it to? There are some things we can learn from other operating systems	. 82

Coming

- FORTRAN: The Forgotten Language
- SPAWN And SPOOL In BP2
- More DCL
- Fitting Files On Floppies (RT11 Directories On Floppy Disks)
- Managing The New RSTS Micro-Spooler
- MAXI In Your MINI
- LDC Software Upgrade RSTS/E V.8.0
- Using The LA100 On A VT100 Printer Port
- System Stats In Assembler (How To Do SYS CALL In Assember?)
- More . . .

Departments

From the Publishers 4
From the VAX Editor 21
"Pro" Talk (letters to the editor) 6
Tips and Techniques (returns next issue)
Dear VAX/RSTS Man (questions & answers) 37
News Releases 86
Classifieds 95
List of Advertisers 96

The VAX/RSTS Professional Magazine, December 1, 1983, Vol. S, No. 6. ISSN 0746-1909 is published bi-monthly by M Systems, Inc., 161 E. Hunting Park Avenue, Philadelphia, PA 19124. Subscriptions: single copy price *10∞, *35∞ per year: \$50 Canada and 1st class, All other countries, air mail, \$60 US. Second Class postage paid at Philadelphia, PA. POSTMASTER: Send all correspondence and address changes to: VAX/RSTS Professional, P.O. Box 361, Ft. Washington, Pa. 19034-0361, telephone (215) 542-7008. COPYRIGHT © 1983 by M Systems, Inc. All rights reserved. No part of this publication may be reproduced in any form without written permission from the publisher.

From the publishers . .

THE MORE THINGS CHANGE. THE MORE THEY ARE THE SAME

Carl B. Marbach

It has been four years since we publishthe first RSTS PROFESSIONAL in November/December 1979. It was an idea whose time had come, and we (I include all our readers and friends) made it happen. Subscriptions grew, advertising filled its pages and we printed more useful articles for DEC computers than had ever been printed before.

Soon, four times a year wasn't enough and we moved to produce the magazine every other month, six times a year. Most of our readers were RSTS users, but at heart we were all DEC interactive computer people. Computers and people don't stand still, and technology decreed that we move along, or up. We struggled with the idea and finally brought out the VAX/RSTS PROFES-SIONAL this year. We worried a lot. Would RSTS people not like us any more? Would VAX people learn to love the "PRO" as thousands of RSTS user had? Would we be on the fence and have no audience?

I can report to you that most of our worries were unfounded. RSTS people are happy to hear about what is happening in the VAX world. There continues to be plenty of good solid technical help for RSTS people and we are still getting new subscriptions from, yes it is true, "NEW" RSTS users.

Thanks to Al Cini, one of the real VAX Professionals, the VAX/VMS persons are finding lots of quality VAX information in our pages. What we are finding is that a lot of material is not so system specific that it is wasted on any one operating system. Is it so far fetched to believe that some VAXs might have MICRO-11s as data collectors or remote systems? Using and knowing about smaller PDP-11s could be important to even the largest VAX user.

Under Al's direction you can count on the VAX articles to keep meeting and exceeding your expectations; of course, he needs your help. If you have something to say about VAX/VMS, do your computing friends a favor and put them on tape or floppy and send them to us. This is the best place to get the message to VAX users all over the world.

New VAX subscriptions are setting records. We have signed up more new subscribers in the last two months than any two month period in our history. It seems that VMS is a poplular operating system. Did you know that our magazine has about four times the circulation of any DECUS newsletter? The VAX/RSTS Professional is read in more than 15 countries around the world. You should see our post office bill!

The VAX/RSTS Professional was conceived to be YOUR technical journal and forum. It is getting bigger (note the new bindery process and increase in the number of pages) and we hope, better. In the computer business four years is a long time but my message to you is the same as it was in 1979: Thank you for helping to produce the only operating system specific magazine in the world. I look forward to the next four years of sharing it with you. Thanks.

OBUS FEVER Dave Mallery

I have to hand it to someone (or several) on the DECwest team that developed the MicroVAX. Choosing the Micro-11 Qbus format was a decision that will pay off for years to come.

The choice has made a VAX a possibility for small system integrators. The whole world of Qbus peripherals is now available. This is getting back to the original spirit of the PDP-11s and the Unibus. Make the bus specs public and let all the world have at it! There are probably more PDP-11s than anything else today because of that fact. DEC may be big, but the add-on industry that they spawned is also big. Since that add-on industry exists to "add-on" to DEC processors, DEC wins too.

There was lots of talk on the exhibition floor at DECUS about Qbus performance. It seems that with the newer block mode DMA controllers, Qbus performance can sometimes exceed Unibus. If this is true, then the last of the "problems" with the bus

I hope DEC can cope with the demand that is going to hit them for this new machine. DEC's roots are in "pumping iron" out the door. These fantastic, cost effective micro boxes will allow them to get back to the good old days.

We'd like to extend to you all our best wishes for a Healthy & Happy 1984!



Publishers: R.D. Mallery, Carl B. Marbach

Managing Editor: James L. Trichon

VAX Editor: Al Cini

Director of Advertising: Helen B. Marbach

Business Manager: Peg Leiby

Advertising Sales Manager: Hope Makransky

Assistant Editor: Linda DiBiasio

Production Manager: Georgia F. Conrad

Subscription Fulfillment:

Kathi B. Campione, Claire Hollister, Margie Pitrone, Connie Mahon, Terry McMahon

Contributors:

Ilan Caron, Alton O. Moore, III, Michael Mowat, John D. Papadopoullos, Robert B. Robbins, Ann Sangpan, Terry C. Shannon, Marci Shatzman Bob Stanley, Allen Watson

United Kingdom Representative:

Pauline Noakes

RTZ Computer Services Ltd., P.O. Box 19 1 Redcliff Street, Bristol, BS99-7JS Phone: Bristol 24181

Mid-Atlantic and Northeastern Representative:

Fd Shaud

P.O. Box 187, Radnor, PA 19087 Phone: 215/688-7233

Cartoons:

Emily Dahlstrom, John's

Design, Typesetting & Layout: Grossman Graphics

Photography: Al Ewing

Printing & Binding: Schneider Litho Co., Inc.

Cover Art: Jim Lang

ALL PROGRAMS PUBLISHED IN THE VAX/RSTS PROFESSIONAL ARE WARRANTEED TO PERFORM NO USEFUL FUNCTION. THEY ARE GUARANTEED TO CONTAIN BUGS. THEY ARE DESIGNED TO GET YOU THINKING. THEY ARE INTENDED TO EDUCATE AND ENTERTAIN. THEY ARE PUBLISHED ON THE PREMISE THAT IT IS BETTER TO SPREAD PEOPLES' BEST EFFORTS AROUND EVEN IF THERE IS AN OCCASIONAL PROBLEM. IF YOU USE THEM, MAKE THEM YOUR OWN, AND YOU WILL NOT GO WRONG.

Editorial Information: We will consider for publication all submitted manuscripts and photographs, and welcome your articles, photographs and suggestions. All material will be treated with care, although we cannot be responsible for loss or damage. (Any payment for use of material will be made only upon

*This magazine is not sponsored or approved by or connected in any way with Digital Equipment Corporation. "VAX", "RSTS" and "DEC" are registered trademarks of Digital Equipment Corpora-tion. Digital Equipment Corporation is the owner of the trademarks "VAX", "RSTS" and "DEC" and is the source of all "DEC" products--

Material presented in this publication in no way reflects the specifications or policies of Digital Equipment Corporation. All materials presented are believed accurate, but we cannot assume responsibility for their accuracy or application.

SYSTEM PERFORMANCE TEAM

Technically Superior Products from System Performance House, Inc.

1. DOPTER

The best disk structuring/optimizing program available. DOPTER automatically performs every optimization available in any other program, plus many more which are found only in DOPTER. DOPTER is available for both version 7 and 8 of RSTS/E. DOPTER is an excellent tool for converting version 7 disks to version 8 type disks. Single CPU price - \$750.00.

2. UFDOPT

A fast, safe REORDR-like program which builds better directories which stay that way longer than those produced by REORDR or RDR*. Do not be confused-only UFDOPT puts the most frequently accessed files at the front of the UFD's, deletes unused attributes, sets the contiguous bit where appropriate, and puts free space near the front of the UFD to speed file creation and to help prevent "tangling" of the directory links. Single CPU price - \$250.00.

3. SPSORT

An extremely fast, versatile RSTS/E sort utility. SPSORT needs no scratch space except for a 5 block control file. It comes in three forms, a SORTG/SORTM* emulator, a Sort-1 Plus* emulator and an FSORT3* emulator. These sorts provide a direct repacement for their namesakes-with no change in software. In addition, SPSORT's emulators have been extended to sort DIBOL decimal, packed and EBCDIC data types. Single CPU price - SORTG/SORTM emulator, \$450.00, Sort-1 Plus emulator - \$850.00, FSORT3 emulator - \$1250.00.

Coming Soon

A single-disk DOPTER which will optimize single-disk systems. There will be no disk packs or tapes to shuffle; just run it and enjoy the performance increase!

For more information on any of the above products, write or call the SYSTEM PERFORMANCE HOUSE, INC.

*SORTM and SORTG are registered trademarks of Digital Equipment Corporation. Sort-1 Plus is a registered trademark of Oregon Software, Inc. FSORT3 is a registered trademark of Evans, Griffiths and Hart, Inc. RDR is a registered trademark of Software Techniques, Inc.



System Performance House, Inc.

Please send	more informatio	n:	
□DOPTER	□UFDOPT	□SPSORT	
Name			
Phone			
~			

System Performance House, Inc.

5522 Loch More Ct. E. Dublin, Ohio 43017 · 614/764-1585



Send letters to:
PRO TALK
P.O. Box 361
Ft. Washington, PA
19034-0361

In response to Fred Gervasoni's letter in the April, 1983 issue of RSTS Professional (Page 71), I would first of all like to answer our young reader's request by explaining how the keyboard number is extracted, and secondly, share this useful piece of code with all the other interested parties. The code kills two birds with one stone and is self explanatory.

K.K.N.G.H. stands for King Khalid National Guard Hospital. The hospital's computer department is manned with professionals from the United Kingdom and is rated as having one of the most advanced technological computer departments in the world.

Thanking you for your time and effort on this matter.

John D. Papadopoullos Senior Analyst Programmer K.K.N.G.H. Jeddah (K.S.A.)

See "The All Singing, All Dancing Skeleton Program" in this issue.

My article, "4th Generation System Development," August, 1983, Vol. 5, No. 4, demonstrates that wrong version problems are not limited to programs!

The version published lacked the statistics which were shown as XXXX rather than the actual numbers. My apologies, the correct version is reproduced below.

"SCOPE

The resulting system is composed of 25 programs distributed between five menus, three master files, one control file, four separate transaction type programs, one major inquiry program, nine selective report/inquiry programs and two extract/merge programs. The system contains 5800 lines of source code and 900 lines of documentation produced by ACT4.

ACT4 STATISTICS

Total ACT4 usage statistics were 322 minutes of connect time and 522 seconds of CPU usage."

Two references that were dropped from the article might be of interest. EDP ANALYSER of March 1981 provides an excellent overview of this subject. ACT4 is located 347 Bay Street, Suite 603, Toronto Ont M5H 2R7.

DG Burnley

I just thought that you might like to know that nobody ever came back to me with an answer as to why TECO/VT performs a wrap around after column 78.

The general view is that this is not caused by VTEDIT macro itself, but by something in the TECO scope driver.

Question: If TECO is so wonderful, why can't it check my spelin for me?

I wonder if you could consider having a competition in the DEC or RSTS PROFESSIONAL, where there would be two prizes.

The first prize would be for the correct answer, while the other prize would be for the most original!

Anyway, keep taking the pills — They appear to be working!

Peter Dick London, England

Regarding the article by Kevin P. Herbert, a VAX-11 BASIC subroutine can not receive parameters by value; it will not even compile! But there is a way around this by using the LOC function. You simply take the LOC of the parameter, thus giving you the value.

WELL, NOW WE CAN FAIL TO COMMUNICATE WITH MORE PEOPLE — FASTER!!



> Allen Rueter St. Louis, MO

Mr. Herbert's article appeared in August 1983, Vol. 5, No. 4 ("Writing AST Routines in VAX-11 BASIC"). A correction to this problem was published in October, 1983, Vol. 5 No. 5 on page 22. However, it bears mentioning again.

EMULEX TALKS DEC



REEL (TO REEL) ALTERNATIVE ...

The new Emulex TC7000 tape coupler is more than just another emulation of DEC peripheral hardware for VAX-11/750 or 780 systems. Like every Emulex disk, tape or data communications controller, it gives you performance and cost advantages that no other product can match. Re-marketing or using a VAX-11/750 system? The TC7000 coupler allows you to move up to the higher densities and transfer rates of state-of-the-art GCR tape transports. Backing up a VAX-11/780 system? The same TC7000 gives you a choice of three tape densities — 800-bpi NRZ, 1600-bpi PE or 6250-bpi GCR. In any combination. In a single tape subsystem, connected to a single plug-in board.

FLICK OF A SWITCH ...

Think what this means in terms of inventory, spare parts, service and system-upgrade possibilities. With the flick of a switch, you can convert a VAX-11/750 tape subsystem to a VAX-11/780 storage facility. Flick another switch and you can convert from one industry-standard tape interface to another. The Emulex TC7000 supports up to four STC-type tape transports or up to eight reel-to-reel units with Pertec-type formatter interfaces.

TRANSPARENT SOFTWARE, COMPATIBLE HARDWARE...

All this without touching the balance of your software or hardware. Your VMS or UNIX operating system just assumes that another TM03/TU77 tape facility has been added to the system. Requires only one Massbus controller slot in a VAX-11/750 backplane. No extra backplane connection at all in the case of VAX-11/780 CPUs. The TC7000 fits right into the Emulex V-Master card cage you have already plugged into your VAX-11/780 terminator slot.

ATTENTION, ALL LSI-11 USERS...

And here's good news for LSI-11 users. With your support, Emulex is now one of the country's leading suppliers of DEC-compatible hardware. With production soaring, it's time to pass along a part of the savings in production costs by reducing prices across a broad spectrum of LSI-11 disk and tape products. Price slashed 33% on the popular TC02 tape coupler. Additional savings of 21% to 32% on UC01, SC02 and SC03 controllers. Phone your nearest Emulex representative for details—and watch this space for even more good news from Emulex.

FROM THE EMULEX FILE...

It's official. Fiscal 1983 was a banner year for Emulex, its customers and its suppliers. A 93% increase in sales. A 155% increase in earnings—much of it to be reinvested in new-product research and new production facilities that will further reduce the cost of Emulex products. To get the full story, write for your copy of the Emulex 1983 Annual Report. It's your story, too.



3545 Harbor Blvd., P.O. Box 6725 Costa Mesa, California 92626 Toll-Free (800) 854-7112, in Calif. (714) 662-5600.

VAX/RSTS SECURITY REVISITED

By Dave Mallery Photo by Philip G. Anthony

I intend to treat five major aspects of security common to both RSTS and VMS installations in this article:

- 1) Physical your ability to keep unauthorized people away from your computer and related communications equipment
- 2) Internal password and other mechanisms either inside or outside the operating system or applications that ensure that the programs are used by the right people at the right times and locations
- 3) Dial-up and Leased line the weakest link in the chain.
- 4) Software operating system bugs and "holes" as well as "Trojan Horses"
 - 5) Psychological defense against "around-end" attack.

PHYSICAL SECURITY

Like any defense, you are only as secure as your weakest link. The press is full of examples of hair-raising penetrations. The systems penetrated are, in many cases, the most physically secure in the world, yet their defenses

seem to peel back with almost no effort.

The reasons for physical security are obvious. If I want to bomb your computer. I have to get the bomb into the computer room or the immediate vicinity. If I want to destroy your data, I have to get into your vault with a magnet. (I hadn't been six months in the business (1967) when a disgruntled employee erased a year of summary data for my employer just prior to a union walkout.)

There are many more reasons for keeping people out of the computer room. All sorts of information about your operation is lying around, coming off the printers, taped to the terminals or written on the

walls. Often ignored are the telephone numbers of your dial in lines that are written on the RJ-11 blocks on the wall, or taped to the modems or cables. If there are terminals other than the console, they are often bereft of security, since they are in an area considered secure. The operator console is a multi-faceted threat. Do you know what is done with the hard-copy that comes off it? Is it logged in while unattended? Can you see the terminal from the computer room window. Can you read the printing?

More "KBO:" horrors: Most security systems and operating systems impart special privileges to the console. For instance, the RSTS system start-up procedure logs the console into [1,2]. If the operator aborts the start-up once INIT.BAC is running, then the console remains logged into [1,2]. Normally, one relies on the physical security of the computer room as a guard against this necessary "hole." There are modifications one can make to INIT, but these render the system vulnerable to a full "lock-out."

This vulnerability does not seem to exist in the VMS world. Provided no one has "tinkered" with the startup.com files, the start-up process is un-interruptable and does not

leave any terminals logged in

It is a good idea to take the "vip" tour of the computer room (the one the boss gives to visiting dignitaries). See how much information you can gather about the system.

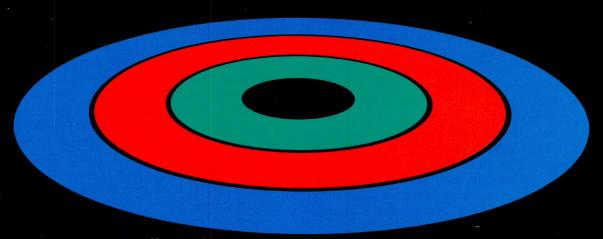
Physical security does not end with the computer room. Tape libraries, staging areas and their terminals, and most important, telephone rooms are vulnerable. Vital data lines are available at several places in your building, perhaps you don't even know where. Remember that the person who wants to crack your system is probably more knowledgeable about the phone system than most phone employees.



WHAT YOU DON'T KNOW **ABOUT RSTS/E 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 wellordered 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.

- 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.

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

Once a phone line has left your "secure" building, it travels to the local central office along with all its peers, in large cables. These cables terminate from time to time up on poles. They are jumped to the next cable and so finally get to the office. They are vulnerable to tapping at any point where they exit a cable. Unless the data is encrypted, it can be displayed with little or no effort, and the tap cannot be detected except by the most sophisticated equipment. The data that is passed on synchronous lines between stat muxs is harder to separate and read, but it is all there. Wouldn't it be a nice feature for stat mux makers to offer . . .

One of the most "famous" tap jobs was accomplished with a little knowledge of phone company internals. The individual called the repair facility, impersonating a phone employee (see psychological attacks). He arranged for a simple bridge arrangement that caused a phone in his room to ring in parallel with a certain dial-in line on a target computer. All he had to do when the phone rang, was to lift the receiver and place it into his acoustic coupler. A verbatim transcript of the entire session was printed out for him, including all the passwords typed, etc.

I was in a computer room recently that was a perfect example. Built in a modern warehouse building, the computer room was equipped with two fine windows that faced the outside world. The windows were single pane glass. The room itself had a motion detector, but as I noted, the alarm would be activated by the bomb as it came in the window. This firm happens to turn over 50 percent of its employees a year, but does not percieve any threat from disgruntled employees. The computer room had an outer room for printing and job staging. That room opens into a corridor (in a building with nearly a hundred employees). Both doors are kept unlocked. Tapes are kept in a vault in the programmer area across the hall. The vault is unlocked and unattended during the day. There is no alarm interlock to insure that the vault is at least closed when the building is locked. No offsite storage of critical backups is performed.

Another area of concern is the physical security of printed forms. Where are your payroll and a/p checks kept right now? Do you even know?

INTERNAL SECURITY

By internal security, I mean the password and other mechanisms that are designed to limit access to the proper people at the right times and places. This can be as simple as the password mechanism, or as complicated as custom software can make it.

There is usually some division of tasks discernable in the physical layout of a company. One would not expect to see payroll running on a terminal installed on a salesman's desk. Nor would one expect to see the general ledger being run by the payroll clerk. Least of all, would one expect to see a warehouseman running the president's private spread sheets. This reality has led to the creation of security superstructures that are based on keyboard-ids. A simple system along these lines can be added to the RSTS login, or included in the VMS login.com procedure.

Passwords in a commercial environment are often a

joke. They are written on terminals, left in the VT100 answerback buffer, yelled across rooms. Terminals are left logged in while the users are out to lunch, and left in the same state all night (to let the cleaning service run accounts payable). How many payroll systems have "PAY" or "PAYROL" as the password, etc... How many RSTS [1,2]s are there out there with "SYSLIB" as the password. Nuff said.

Another word about passwords. They should be:

- 1) full allowable length
- 2) contain at least one numeric digit
- 3) not be related to the owner or the application

Secondary passwords are a great idea. They can be requested by the login procedure, or by the application.

Another worry is that valid users, once validly logged in, will proceed to exit their legal "space" and go elsewhere in the computer, doing things they oughtn't. This can be countered by the above mentioned keyboard sensitive technique: KB43: is allowed into [7,3] from Monday to Friday and from 8:45 to 17:30 only. Any time this keyboard is found elsewhere, or outside the time range, the alarm is broadcast to the appropriate parties.

Leaving a trail is critical. There are a number of messages produced by the RSTS login program that are broadcast to KBO:. Before you patch these "annoyances" out, realize that they may consititute the only warning you may ever get. Any security system you buy or implement should have a full and flexible alarm system. Alarms on an unattended KBO: at three in the morning are worthless if they are never read . . .

DIAL-UP and LEASED LINE

Allowing dial-up access to your computer is probably a necessary evil. Only huge companies can afford 24-hour, on site programmer support.

Once you have plugged that little RJ-11 plug into the box on the wall, you have exposed your computer to the world-wide switched network . . . the world of the hacker. Probably the only thing worse than that is to plug your computer into one of the packet networks, or the ARPANET.

The popular press is full of lurid accounts of how it is done. Your must convince yourself that probes from the network are a certainty, and that they are going on right now. If this is the case, do you know that they are not?

There are as many remedies as methods of birth control. They vary from abstinence to high-tech devices. Let's look at abstinence. This is a very inexpensive yet highly effective method. You unplug the phone line from the modem. You set up a procedure with the operator that guarantees that the plug will only be inserted after a specific identification sequence is performed that includes a call back by the operator to a known phone number not supplied by the caller.

That is great if you have 24-hour coverage. And if the

Start with an LSI-11

Dataram's W23 starts with the LSI-11/23 I/O mapping accommodates (soon the LSI-11/73) 1.0 MB minimum... 18-bit peripherals (e.g. RX02) up to 4.0 MB! and 22-bit peripherals, while supporting 4.0 MB of memory 27 dual-slot backplane RM02-compatible 80 MB or 160 MB Fujitsu Winchester W23's RM02-compatible controller can support external CDC 9762 SMD drive for removable media capability or backup, in addition to controlling the integral Winchester \$16,940 system price* Supports RSX-11M, RSX-11M-PLUS, ...and end up with a RSTS, and UNIX powerful minicomputer!-Dataram's W23 minicomputer system gives you more than an LSI-11. By packaging an LSI-11/23 (or LSI-11/73) in a compact 101/2" chassis with 160 MB Cipher Microstreamer with controller Winchester capacity, 27 dual slots, and a hefty 50 A power supply. And then going a step further with I/O mapping that allows up to 4.0 MB of memory while available for less than \$5300 supporting both 18-bit and 22-bit controllers. For more information, forward this coupon to us or, even better, call us today at (609) 799-0071. ☐ Send information. □ Contact me immediately. Name Company Address State Zip Phone * \$16,940 system price is single-quantity domestic price for W23 with LSI 11/23, 1.0 MB, 80 MB Winchester, and I/O mapping. Return to:

DATARAM

DEC, LSI-11, RSTS, and RSX are trademarks of Digital Equipment Corporation. UNIX is a trademark of Bell Laboratories.

Dataram Corporation, Princeton Road, Cranbury, NJ 08512

operator remembers to unplug the modem when you are done. This could also be accomplished by the security personel in larger operations.

RSTS has an option for a secondary password on dial-in lines. You can never have enough passwords!

Security packages exist that provide a wealth of protection for dial-in lines including probe detection and automatic line disabling after a specified number of attempts.

A newer, but very expensive technique is a variant of the answering machine that answers the phone, demands a touch-tone password, and then calls back a pre-stored number to make the connection. For the shop with only a few dial-in lines, this is nearly optimum. (Remember that all of these techniques are vulnerable to the tap technique . . . they can tap the call-back number . . .)

Everything said about the dial network should be raised a few orders of magnitude when discussing the packet networks.

Leased lines imply that terminals connected directly to the computer are located in remote locations. Not only can the line be tapped (with more difficulty), but there are a number of directly connected terminals in a remote location that is probably not as secure as your own location.

A super-simple expedient is to have a procedure that sets all the line speeds to zero outside of business hours. It is very difficult to log in at speed zero. This is worth the occasional heat you will take in emergency situations. The same can be said of in-house terminals after hours.

SOFTWARE AND "HORSES"

When I was a kid (never mind when!), there was a chant that went: "There's a hole in the bottom of the sea . . ."

Face it: there are hundreds of thousands of lines of code in both RSTS and VMS. Have you ever seen a program of more than five lines that didn't have a bug????

Fortunately, these "programs" do get a great deal of use. Occasional holes do appear and are plugged with great speed. This is due in no small part to the current crisis of security. The best way to keep up on the current status of your favorite operating system is to attend the DECUS product panels. Many break-in artists do.

The "Trojan Horse" is a different beast. It is possible for the author of a third party (or DIGITAL) software product to leave a "back door" in a piece of software. This will be activated sometime in the future by a convoluted series of normal commands, or a special hidden command. The effect of the activation can be anything from entry to privilege to total destruction of the system. I feel that you should be aware of this possibility and bring it up in the negotiations for any software purchase. There is really nothing that the vendor can do, but the possibility is too real.

PSYCOLOGICAL ATTACK

Perhaps the most successful break-ins have occurred not by a frontal attack on the physical or software defenses of an installation, but by a script that goes like this: TIME: 2:15AM

PLACE: THE COMPUTER ROOM

(phone rings)

OPER: Hi — computer room XYZ Corporation, Suzy

speaking . . .

ENEMY: Oh, Hi Suzy, boy am I glad I got through to you . . . This is DEC software support. Did you get the emergency patch we sent over earlier?

OPER: NO! What patch . . .?

ENEMY: You didn't . . . this is horrible . . . Is the machine still up??

OPER: yes . . .

ENEMY: Phew! we're in luck! I will have to give it

to you over the phone . . OK?

OPER: Sure

ENEMY: OK, Suzy, just log your console in and run \$ONLPAT . . . Don't worry, I'll do it very

carefully with you . . .

The net effect of this little scenario is the penetration or destruction of the system after the next boot.

You can't expect your employees to think of every possibility. A computer room needs a certain line of defense against this kind of "around-end" penetration. You must set up procedures for incoming phone calls at night (and at all times!). You should brief operators on this technique. They should know who is calling them, and should never accept instructions over the phone without at least a call-back to a known phone number.

WHAT TO DO WITH A PENETRATED SYSTEM

OK, your system has been broken. Now what?

First, you should consider the possibilities of catching the criminal. Most of these cases go unreported because the owners of the business don't want to lose face. The FBI would love to help. You should also have a talk with DEC Security. You should do that before you change anything.

Once you decide to clean up the situation, here is a list of things to do:

- 1) Save all suspect media as evidence
- 2) Do a cold-start sysgen on empty media using secure copies of the distribution. If you think your distribution is suspect, get a new one.
 - 3) Have an outsider do the sysgen.
 - 4) Replace all utilities and third party software.
- 5) Re-compile any programs on the system that have any level of privilege. It's a good idea to periodically recompile privileged user programs from secure sources without notice.
- 6) If you know who did it, swallow your pride and prosecute! We will all be in your debt.

THERE ARE AT LEAST EIGHT MORE TERMINALS HIDING IN YOUR VAX.

All you need is one Able VMZ/32 asynchronous terminal controller to find 8 more lines in your DEC VAX-11/730, /750 or /780, for a total of 16. With an additional VMZ/32 you can discover 16 more lines.

The VMZ/32 board plugs into the same space as the standard DEC DMF32 board it replaces, providing 16 lines instead of 8. Without patches to the VMS. And with complete VMS compatibility.

Immediately you'll unleash a minimum 13% increase in processing speed. You'll instantly realize a 50% reduction in costs per line, obtain a 50% space savings in the backplane and see a surprising reduction in power consumption.

The VMZ/32 also reveals hidden performance within your CPU by increasing its availability (idle time) up to 500%.

To handle this newly acquired power, the VMZ/32

gives you modem control on all sixteen lines. This lets you easily manage the terminals, including those remotely located. Plus, you can easily vary the speed on each line from a 50 to a 19.2K baud rate, including split baud capability.

Here's a dependability clue: an MTBF of over 100,000 hours. To back it up, the VMZ/32 comes with a one-year warranty from Able that can be extended if you like. Ask your Able representative for full details.

If you do unearth a problem, Able Support Service is as close as the phone.

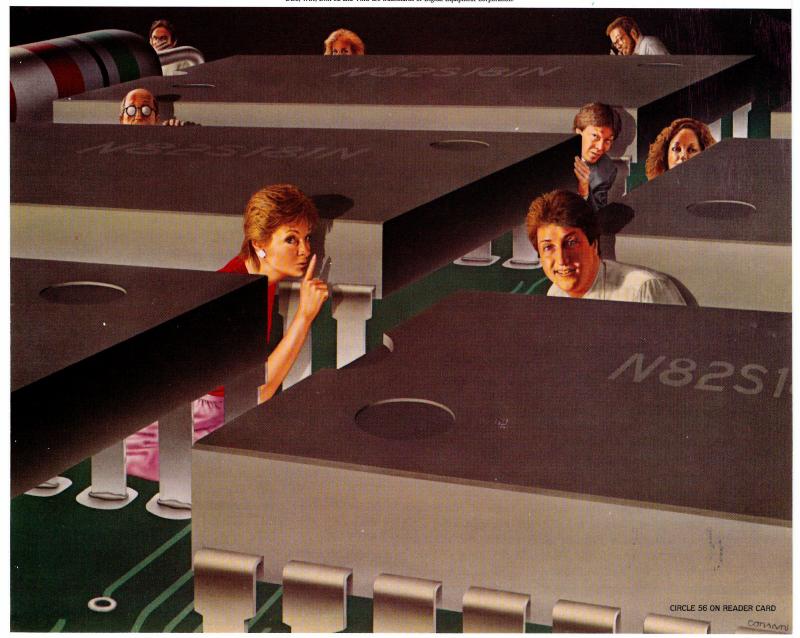
You can also uncover extra performance hidden in your printer with Able's VMZ/LP. The VMZ/LP provides a dramatic reduction in CPU overhead and is LP11 cable compatible.

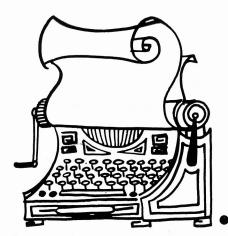
So, to discover the full potential hiding in your VAX, Able's got a simple solution: just plug in the VMZ/32.

The communications specialists.

1732 Reynolds Avenue, Irvine, California 92714. Call toll free: 800-332-2253. In the Irvine area: (714) 979-7030. Or, TWX: 910-595-1729

DEC, VAX, DMF32 and VMS are trademarks of Digital Equipment Corporation.





A FEW NOTES FOR INFORMATION ARCHITECTS

By Al Cini



Wrap a comprehensive assortment of languages, utilities and data base software around a central dictionary and VOILA! Information Architecture! (VAX Information Architecture, to be specific, or VIA for short).

To be a real Information Architect, though, you'll have to commit yourself to some planning, organizing and standardizing. This article will briefly review some VIA fundamentals and relate some tips not covered in the manuals that will, if not make a Frank Lloyd Wright out of you, at least advance you to "stone mason" level.

UNDERSTANDING VIA

VAX Information Architecture is a collection of separately purchased and independently useable software products which "play together" by way of the Common Data Dictionary (CDD) facility. While you can buy a database system (DBMS-32) that works within the VIA framework, you certainly DON'T need it to get the major benefits of a "data base management" approach to developing software. The CDD, Datatrieve, a programming language, and a little mental discipline are really all you need to accelerate or often eliminate a lot of program development and maintenance. Some of this "discipline" will surface during the following "walk-through" examples. We'll also see some of the VIA products at work, but you should (of course) refer to the documentation for specific functional details.

IN GENERAL

Who will be your Data Base Administrator?

DBA, like System Manager, is a very important part-time job in any programming shop, even single-programmer shops. The DBA is responsible for collecting and defining the user community's data requirements, and making sure that the Common Data Dictionary really is the ONLY central dictionary facility in use at the installation. When all software intersects through the CDD, the CDD becomes a de facto "design document" that can help make programmers more "interchangeable." Thoughtful maintenance of the Common Data Dictionary is a very important job, without which the "Architecture" is reduced to a mere drafting tool. So:

Appoint someone or yourself as "Data Base Administrator," whether you bought DBMS-32 or not, and make sure the job is taken seriously.

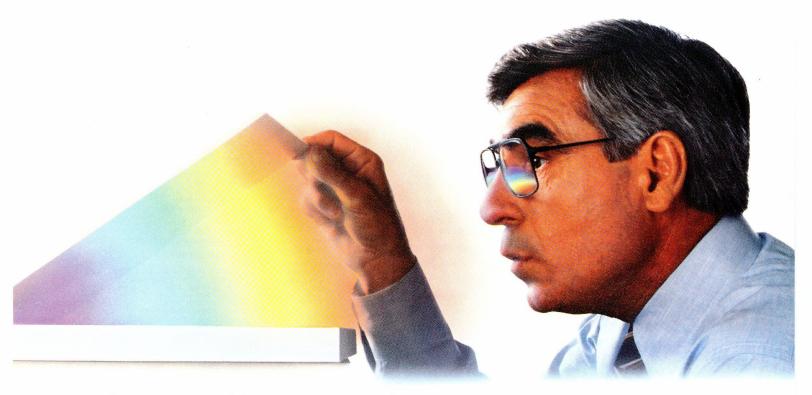
THE CDD DIRECTORY

Directories in the Common Data Dictionary organize objects, such as record definitions, DTR domains, etc., in a hierarchical scheme, downward from the CDD\$TOP entrypoint node. As a user, you are free to organize this directory any which way you like (Chapter 1 of the CDD Utilities Reference Manual describes the directory structure and suggests some organizations), but a few arrangements just won't "work" very well at all:

- 1. It's not a good idea to separate different kinds of CDD objects into different "trees" of the CDD directory. This is no big deal for some products in VIA, but if you're using DATATRIEVE and your domains, records, procedures, views, etc. are arranged along different directory paths, you'll go crazy trying to work with them. Since each CDD object will have a "type" associated with it (e.g., CDD\$RECORD) and since DMU allows you to qualify some directory management commands with a /TYPE qualifier, separating objects by type within the directory is rarely necessary.
- 2. Avoid naming CDD objects with a language's reserved or illegal words. CDDL won't mind compiling an object description named "GOTO" but %INCLUDing or COPYing it into a program later can cause headaches. Remember that, since new VIA products may be bought further "down the line," you may need to anticipate them somewhat when you . . .
- 3. Establish some sensible naming conventions for objects and object descriptions. In this area, just a little thought can go a long way. Ending DTR domain, record, and view definitions with a "type suffix," such as ORDER_REC and ORDER_DMN, will allow you to use the same generic name (ORDER) for both objects. Further, since CDD objects are not backed up by versions the way files are, you might consider a version suffix (ORDER_DMN_V1) as well. These can help you when you use DATATRIEVE to restructure a changed file, or to adopt the kinds of "run-time" binding techniques using Shareable Images we discussed in the last issue. These kinds of "standards," such as making sure all variable names have an underscore in them somewhere, can also help you avoid "colliding" with a language reserved word later.

For our example, we'll be referring to CDD objects defined in CDD\$TOP.ORDERS. To simplify working in this

The Newest Innovation in Controller Technology From the Oldest Name in Multifunction Controllers.



Introducing the World's First LSI-II Emulating Multifunction Disk/Tape Controller.

Say hello to SPECTRA 25, the cornerstone of our new family of high-performance disk/tape controllers designed for use with DEC's LSI-II computer.

This revolutionary Q-Bus compatible single quad board lets you interface any combination of two SMD disks and four formatted $\frac{1}{2}$ -inch tape drives. By using extended commands to program the onboard E^2PROM , you can easily select drive mixing, mapping, and many other features—all without removing the controller from the system.

The SPECTRA 25 emulates DEC's RM02/5 and RM80 disk subsystems, and DEC's TSII tape subsystem. It also provides complete emulation for operation with DEC's RT-II, RSX-IIM, RSX-IIM-PLUS and RSTS/E operating systems.

To further enhance system performance, Spectra Logic offers SPECTRA STREAM™ software, a streaming tape backup utility that can back up an entire 80MB drive in only seven minutes.

Spectra Logic first introduced the multifunction concept back in 1979. And we've been quietly revolutionizing the market ever since with families of controllers that provide the high-performance, proven reliability, and added value you need to stay competitive.

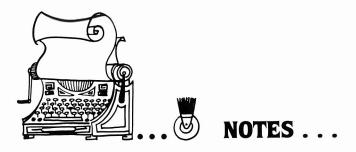
We also offer the industry's most comprehensive one year warranty, responsive nationwide service, and ongoing technical support.

SPECTRA 25 is the latest innovation in controller technology from the company with peripheral vision. Spectra Logic. For further information, including complete technical specifications, call or write us today.

Spectra Logic Corporation 1227 Innsbruck Drive, Sunnyvale, CA 94089 (408) 744-0930 TWX 910-339-9566 TELEX 172524 SPL SUVL

> International Sales Office: Belgium (32) (2) 5134892

SPECTAN LOGIC



directory, we'll set our "default" CDD entrypoint: \$DEFINE CDD\$DEFAULT CDD\$TOP.ORDERS

. . . permitting us "shorthand" access to its objects. To avoid confusion, though, our CDD definition source files will:

Explicitly state the target locations of defined objects, rather than use CDD\$DEFAULT. Particularly within command procedures and .DDL files, this eliminates the uncertainty that CDD\$DEFAULT is set properly before proceeding and prevents objects being scattered all over the CDD by mistake.

DEFINING CDD OBJECTS

The CDD shouldn't be something you wake up in the morning deciding to use. The natural organization of your application's data should dictate your approach to defining it within the CDD, and not the other way around. As a documentation tool, a CDD definition for a data structure may "cue" a maintenance programmer as to how data is used in the application, even if the CDD entry itself is never directly used in any layered software.

For example, lets start with two simple, related records:

- 1. An order header record describes an order placed by a customer for some products, consisting of:
 - a) An internal order number
 - b) A customer identifier
 - c) The date of the order
 - d) The number of products ordered
- 2. An order line item record describes a particular product within a customer order, using:
 - a) An internal order number
 - b) A stock number
 - c) A description of the item
 - d) A price
 - e) A quantity ordered

Several order line item records will be related to their parent "order" record by way of the internal order number.

Notice first of all that the "internal order number" appears in both record definitions. By the same token, "stock number" is likely to appear in some related stock file (for simplicity's sake, we haven't "normalized" these things completely). So, why define these fields over and over again everywhere we refer to them? Instead:

Define commonly used data structures as generic templates and "COPY" them into the definitions that use them. When you make changes to these templates, the CDDL/RECOMPILE command can be used to rebuild all their referencing definitions.

The definitions for stock and internal order numbers follow:

```
(In ORDERNO.DDL:)
      DEFINE RECORD CDD$TOP.ORDERS.ORDER_NUM_FLD_V1
                         /* EXAMPLE FIELD DEFINITION */.
                ORDER_NUM_V1 STRUCTURE.
                         ORD_PFX DATATYPE IS TEXT
                                           SIZE IS 3 CHARACTERS.
               ORD_NUM DATATYPE IS TEXT
SIZE IS 3 CHARACTERS.
END ORDER_NUM_V1 STRUCTURE.
      END ORDER_NUM_FLD_V1 RECORD.
(In STOCKNO, DAL :)
      DEFINE RECORD CDD$TOP.ORDERS.STOCK_NUM_FLD_V1
               DESCRIPTION IS
                         /* EXAMPLE FIELD DEFINITION */.
                STOCK_NUM_V1 STRUCTURE.
                                           DATATYPE IS TEXT
SIZE IS 3 CHARACTERS.
DATATYPE IS TEXT
                         ST KEY
                                           SIZE IS 4 CHARACTERS.
                         ST SFX
                                           DATATYPE IS TEXT
                                           SIZE IS 1 CHARACTER.
                END STOCK_NUM_V1 STRUCTURE.
      END STOCK NUM FLD_V1 RECORD.
```

Now we can define the order header and line item records:

```
(In ORDERH.DDL:)
         DEFINE RECORD CDD$TOP.ORDERS.ORDER_HEADER_REC_V1
                     DESCRIPTION IS
         /* EXAMPLE RECORD DEFINITION */.

ORDER_HEADER_VI STRUCTURE.

ORDER_NUM COPY CDD$TOP.ORDERS.ORDER_NUM_FLD_VI.
                     DT ORD
                                               DATATYPE IS DATE.
                      CUST_NUM
                                               DATATYPE IS TEXT
SIZE IS 5 CHARACTERS
         LINE_ITEM_CNT DATATTE
END ORDER_HEADER_V1 STRUCTURE.
END ORDER_HEADER_REC_V1 RECORD.
                                               DATATYPE IS SIGNED WORD.
         DEFINE RECORD CDD$TOP.ORDERS.ORDER_LINE_ITEM_REC_V1
         /* EXAMPLE RECORD DEFINITION */.
ORDER_LINE_ITEM_VI STRUCTURE.
ORDER_NUM COPY CDD$TOP.ORDERS.ORDER_NUM_FLD_VI
                                               COPY CODSTOP.ORDERS.STOCK.NUM_FLD_V1.
DATATYPE IS SIGNED WORD
EDIT_STRING FOR DTR IS "9999".
DATATYPE IS PACKED DECIMAL
SIZE IS 5 DIGITS 2 FRACTIONS
                      STOCK_NUM
QTY
                      PRICE
                                               EDIT_STRING FOR DTR IS "$$99.99".
DATATYPE IS TEXT
                     DESCRIPT
                                               SIZE IS 10 CHARACTERS.
         END ORDER_LINE_ITEM_REC_V1 RECORD.
```

We can further employ the "generic template" capability of the CDD to define a "view" of a complete order, consisting of a header record and up to 10 related line items, by using the order header and line item records as generic templates:

```
(In ORVIEW.DDL:)

DEFINE RECORD CDD$TOP.ORDERS.ORDER_VIEW_V1

DESCRIPTION IS

"EXAMPLE "VIEW" RECORD DEFINITION "/.

ORDER_VIEW_V1 STRUCTURE.

ORDER_HEADER COPY CDD$TOP.ORDERS.ORDER_HEADER_REC_V1.

ORDER_DETAIL STRUCTURE OCCURS 10 TIMES.

ORDER_LINE_ITEM

COPY CDD$TOP.ORDERS.ORDER_LINE_ITEM_REC_V1.

END ORDER_DETAIL STRUCTURE.

END ORDER_VIEW_V1 STRUCTURE.

END ORDER_VIEW_V1 STRUCTURE.
```

Once we've inserted these descriptions into the $\ensuremath{\mathsf{CDD}}\xspace$. . .

```
$cddl stockno, orderno, ordern, orview
... we can list them ...
$DMU LIST/FULL
```

ORDERS.ORDER HEADER REC_V1 <CDD\$RECORD>

Created by VAX-11 CDD Data Definition Language Version 2.2 . . .

```
Source:

DEFINE RECORD CDD$TOP.ORDERS.ORDER_HEADER_REC_V1

...

END ORDER_HEADER_REC_V1 RECORD.

Description:

EXAMPLE RECORD DEFINITION

Used as a generic record by AL (UIC [100,1]) in process AL ...

on 2-NOV-1983 16:14:27.38.

Explanation:

CDD$TOP.ORDERS.ORDER_VIEW_V1
```

IS COBOL OBSOLETE? PRO-IV USERS THINK SO.

A totally new approach to application development.

Now you can create fully operational applications in days instead of weeks... and without any programming at all!

PRO-IV is a complete application development and execution system.

Creates applications that are fully transportable between VMS, P/OS, RSTS and RSX*

Generates documentation automatically

A complete library of application packages is available.

For more information, call: (617) 486-9621

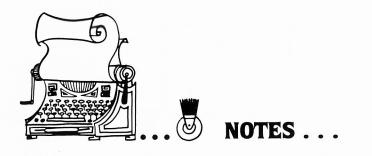


119 Russell Street P.O. Box 595 Littleton, MA 01460

"THE FOURTH GENERATION APPLICATIONS PROCESSOR"

*UNIX available this fall from Pro-IV.
RSTS, RSX, VMS, and P/OS are trademarks of Digital Equipment Corporation.
UNIX is a trademark of Bell Laboratories.

CIRCLE 205 ON READER CARD



... and note that the CDD "history" entries provide us with a cross-reference of generic templates, tracing "who copied whom" and helping us to assess the impact of a change to a crucial field or record definition.

SOME DATATYPE NOTES

Keep in mind that many of the comprehensive datatypes supported by the CDD are not directly supported by some of the layered products. For example, the following VAX-11 BASIC code %INCLUDEs our record definitions:

```
10 $INCLUDE $FROM $CDD "ORDER_HEADER_REC_VI"

$INCLUDE $FROM $CDD "ORDER_LINE_ITEM_REC_VI"

$INCLUDE $FROM $CDD "ORDER_VIEW_VI"

END
```

When we compile it . . .

\$BAS/AUDIT/LIST CDDSAMPLE.BAS

... we get the following gentle wrist-slaps:

We used the CDD "date" datatype to represent "date ordered," but VAX-11 BASIC has no date data type. BASIC "informs" us with a message, and allocates an 8-byte string for the data item, which we can pass to any of the several RTL routines which process dates.

Our CDD "Order View" contains an OCCURS clause, which is considered indexed starting with element 1. Since all BASIC arrays are zero-indexed, an informational message advises us that corrective action has been taken (you may have to specify /SHOW = CDD_DEFINITIONS to CDD objects to print in your BASIC listings):

(The /AUDIT qualifier \$BAS command instructs BASIC to write a record to the CDD whenever it %INCLUDEs anything.)

SOME NOTES ON DATES

Even though the DATE datatype isn't supported by most programming languages, there are many system service and RTL routines which can be used on the 8-byte string area where the internal date is kept, and DATATRIEVE can act directly on dates to do some sophisticated formatting and arithmetic. It pays to use the standard internal VMS date format to store your dates, but watch out for internal dates as segments of string keys in indexed RMS files! They don't "sort" properly, and will appear to be arranged in random order when processed sequentially.

UNDERSTANDING VIEWS

You may be wondering what, beyond its documentation value, our "order view" record offers us. It can't be used as a DATATRIEVE view (we'll see later how they're related, though), and seems worthless within an application program.

Well, if you have the TDMS terminal management package, the "order view" structure can be mapped to a screen format with the order header as a static portion of the display and the order line items mapped into a scrolled region. You can then manage an entire order with a single TDMS request to the abject amazement of your colleagues! We'll go into some detail on TDMS in a future article.

Even if you don't have TDMS, order processing programs written in a higher-level language can process an order stored internally within the "order view" area. To speed things up, the RFAs of records whose data is kept within the view can be retained in an array and used for rapid subsequent update of the disk records.

USING DATATRIEVE WITH THE CDD

Objects are kept in the CDD in their original source form and in an internal form. Consequently, "what you see" is not exactly "what you've got." For example:

```
$DTR
VAX-11 Datatrieve V2.0
DEC Query and Report System
Type HELP for help
DTR> SHOW RECORDS
Records:
                                             ORDER_LINE_ITEM_REC_V1
          ORDER HEADER REC V1
          ORDER_NUM_FLD_V1
                                             ORDER VIEW V1
                                                                  STOCK NUM FLD V1
DTR> SHOW ORDER_HEADER_REC_V1
DEFINE RECORD CDD$TOP.ORDERS.ORDER_HEADER_REC_V1
          DESCRIPTION IS
/* EXAMPLE RECORD DEFINITION */.
ORDER_HEADER_VI STRUCTURE.

ORDER_NUM COPY CDD$TOP.ORDERS.ORDER_NUM_FLD_VI.
                               DATATYPE IS DATE.
DATATYPE IS TEXT
          CUST_NUM
                               SIZE IS 5 CHARACTERS.
DATATYPE IS SIGNED WORD.
LINE_ITEM_CNT DATATY
END ORDER_HEADER_V1 STRUCTURE.
END ORDER_HEADER_REC_V1 RECORD.
```

Notice in the above example that our SHOW command gets us a copy of the order header record in CDDL format. If we extract the record definition into a file, however, we see it translated to DATATRIEVE source format:

```
DTR> EXTRACT ORDER_HEADER_REC_V1 ON SYS$OUTPUT;

DELETE ORDER_HEADER_REC_V1;

DEFINE RECORD ORDER_HEADER_REC_V1

01 ORDER_HEADER_V1.

03 ORDER_NUM.

05 ORD_FFX PIC X(3).

05 ORD_NUM PIC X(3).

03 DT_ORD USAGE DATE.

03 CUST_NUM PIC X(5).

03 LINE_ITEM_CNT USAGE WORD.
```

Note, however, that this DATATRIEVE record has the copied generic templates included directly within it. Hence, another rule of thumb:

Introducing the Most

POWERFUL

General Ledger Software Solution

What is MAPS/GL?

MAPS/GL is a comprehensive general ledger/financial management system designed for DEC's VAX series of minicomputers. General ledger accounting, financial control, budgeting and advanced reporting functions are incorporated in MAPS/GL, and integrated with decision support to form a state-of-the-art financial management software system.

Standard general ledger functions are only the beginning. MAPS/GL includes many advanced features: a powerful, user-friendly report writer; extensive audit trails; three levels of security; capabilities for custom allocations and complex consolidations; and more.

All the Flexibility You'll Ever Need You can tailor MAPS/GL to your individual needs, controlling accounting structures, organizational roll-ups and reporting specifications. As your company grows and your needs change, you can change your system to adapt.

Its powerful but easy-to-use report writer brings flexibility to your daily needs as well. Ad hoc reporting from MAPS/GL's financial database can solve your special information needs.

Truly Interactive Processing

MAPS/GL was designed for today's powerful, interactive minicomputers—not adapted from an out-of-date mainframe-oriented batch system. Inquiry, updating and reporting functions can be performed online, right now. So your accounting staff becomes more productive. And your information much more accessible to management.

Integrated with Decision Support
Best of all, MAPS/GL can be linked
with Ross' easy-to-use packages for
financial modeling and business
graphics, MAPS/MODEL and
MAPS/GRAPH. Together these
packages form an integrated financial

management system, including microcomputer-based distributed computing power with our MAPS/PRO package.

More MAPS from Ross Systems
The MAPS™ (Management, Accounting, and Planning Software) family includes MAPS/AP for accounts payable, MAPS/DB for database management, and MAPS/ISO for employee stock option tracking. All Ross products are available for license or via ROSS/NET™, our VAX-based remote computing service.

And our commitment includes customer and product support that reflects 11 years of consulting experience.

MAPS from Ross Systems. The single source for integrated, interactive financial management software.

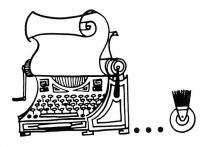
DEC and VAX are trademarks of Digital Equipment Corporation. MAPS and ROSS/NET are trademarks of Ross Systems, Inc.

MAPS / GL was developed by Price Waterhouse under the name "FM80" and adapted for the VAX by Ross Systems, Inc. Our license includes the support necessary to keep MAPS / GL current with new FM80 releases. Price Waterhouse markets and supports FM80 on computers other than VAX.

MAPS Interpretation of the Interpretation of

DISTRICT OFFICES: Palo Alto, CA San Francisco, CA Los Angeles, CA Dallas, TX New York, NY ROSS SYSTEMS CORPORATE HEADQUARTERS 1860 Embarcadero Road Palo Alto, CA 94303 (415) 856-1100

CIRCLE 1 ON READER CARD



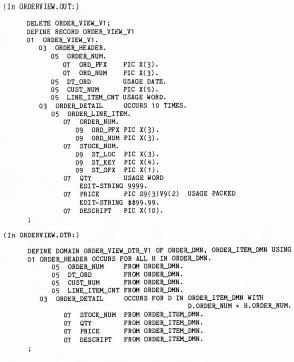
NOTES . .

When you use the generic template record capability of the CDD, NEVER EDIT A CDD DEFINITION USING DATATRIEVE! The COPY clauses will be clobbered and your template advantage lost!

The ability to extract a CDD definition into DTR format does, however, have its good points:

DTR > SHOW DOM	AINS						
Domains: ORDER_DMN ORDER_ITEM_DMN							
DTR> PRINT AL	L ORDER_						
			LINE				
ORD ORD D	-		ITEM				
PFX NUM 0	DRD N	MUI	CNT				
	4000		2				
YBT 123 2-Ja							
YBT 125 2-Ja			3				
YBU 123 12-0c	t-1983 2	?1−G	0				
YBV 123 15-Se	p-1983 2	234	0				
YBV 124 12-No	v-1983 4	1523	0				
DTR> PRINT ALL ORDER_ITEM_DMN;							
ORD ORD ST S	ST ST						
PFX NUM LOC K	EY SFX	QTY	PRICE	DESCRIPT			
mon .	1000 A	0005	*21 50	WIDGET BOX			
YBT 123 TOP 1 YBT 123 TOP 1	1000 A		\$03.00	WIDG. RACK			
YBT 123 TOP 1	1001 A	0000					
			#01 00	FLANGELIPS			
YBT 125 BOT 3				THREADS			
YBT 125 MID 3	300 H	0031	\$00.50	IIIII			
DTR> EXTRACT ORDER_VIEW_V1 ON ORDERVIEW.OUT;							

The translated DTR form of our "order view" record can easily be edited into a DATATRIEVE view:



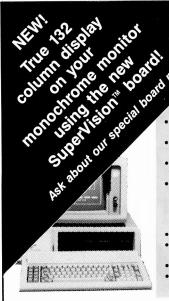
We can use this DTR view as follows:

DTR> @ORDERVIEW.DTR; ! To define the DTR view DTR> READY ORDER_VIEW_DTR_V1 READ DTR > PRINT ALL ORDER_VIEW_DTR_V1 ORD ORD CUST ITEM ST ST PFX NUM ORD NUM LOC KEY SFX QTY PRICE DESCRIPT 2 TOP 1000 YBT 123 2-Jan-1983 21-A A 0008 G 0003 H 0003 TOP 1001 \$03.00 WIDG. RACK \$02.00 FLANGE YBT 125 2-Jan-1983 21-A \$01.00 FLANGELIPS BOT 300 MID 300 H 0031 \$00.50 THREADS YBU 123 12-Oct-1983 21-G

CONCLUSION

YBV 123 15-Sep-1983 234 YBV 124 12-Nov-1983 4523

ARCHITECTURE = THOUGHT + PLANNING + COMMON_SENSE + VIA!!! Write up your own unique experiences with VAX/VMS Information Architecture products and send them in!



Now, join the thousands of large and small companies, universities and individuals who have discovered the advantages of using SMARTERM™/PC VT100 emulator software from Persoft, Inc.

SMARTERM™/PC Emulators Include:

- full emulation of terminal characteristics up to the limits of standard IBM PC* hardware
- 75 to 9600 baud operation with x-on/x-off support
- integral, high-speed, bi-directional, protocolfree ASCII file transfer
- optional error-free protocol which allows ASCII and binary files to be transferred between PC's or between the PC and the host computer (requires user programming on host computer; includes sample VAX programs)
- · multiple "user-friendly" setup configurations user-defined softkeys for simplified auto-dial
- and log-in support
- · full local printer support
- runs under PC DOS operating system using monochrome or 80-column color monitor

- · keytop chart (fits on keyboard), reference card, and user manual
- 30-day money-back guarantee if not satisfied with program
- · unlimited telephone support during normal business hours

Currently available for:

- VT100, VT101, VT102, and VT52 terminals
- · Dasher D100, D200, D400 terminals

- Coming soon:
 VT125 graphics terminal • TeleVideo 950 terminal

We accept company purchase orders, MasterCard, and Visa

*Also runs on Compaq®, Corona®, and Columbia® computer systems.

Available through your local computer dealer or directly from:



2740 Ski Lane Madison, WI 53713 (608) 273-6000

Software for Professionals . . . by Professionals

SMARTERM is a trademark of Persoft, Inc. • SuperVision is a trademark of California Computer Systems, Inc. • VT and VAX are trademarks of Digital Equipment Corporation • Dasher is a trademark of Data General Corporation

From the VAX editor . .

BIG IS SMALL

Al Cini

Some new VAX products attracted a whole lot of attention at October's Las Vegas DECUS, and for good reason: come early next year, multi-user VAX configurations running VMS will cost less than \$30,000 using a new VAX-11/725. For special applications, particularly embedded within OEM products, an even smaller/ cheaper "MicroVAX" will extend 32-bit VAX architecture down into new frontiers of affordability.

But the really hot DECUS crosstalk centered on the recently tumbled value of DEC stock. For those of you who haven't heard, reports of "flat" earnings in their first quarter and worries on Wall Street about slipping DEC management resulted in a drop in the value of their stock of about 30 points over three days (when a presenter mentioned "DECSlide" at one of the sessions, I thought for a minute he was referring to their stock; I was wrong its a VAXbased graphics package). I've always personally believed that Wall Street ought to be moved to Atlantic City and maybe made a property on the Monopoly board, but there's no denying some foundation to the Market's concerns.

Growth is never easy, particularly the kind of breakneck-paced growth Digital has been both blessed and cursed with during its short lifetime. A company will get used to things "coming easy" after a while, start organizing itself into committees, appointing "coordinator-types," establishing task forces, and more or less random-walk its way through its future. When times get tough, though (its called "competition"), shaking off the cobwebs and getting down to business can be a slow and painful process.

But think about those little VAXs. Small, economical, a focused and consistent element of a well-planned and powerful product strategy — maybe in this case DEC can learn something from one of its own products.

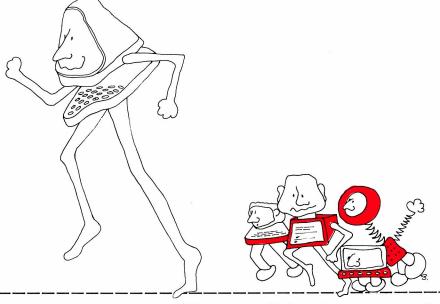
WHAT IS THIS?

HINT: It's Australian!

> **ANSWER NEXT ISSUE**



FOR VAX" RESOURCE ACCOUNTING, PACS™ IS AHEAD OF THE PACK



The competition isn't even close. PACS is so good, it's the only resource accounting package in DEC's™ EAS library. Here's why:

- Flexible billing algorithms
- · User budgeting and port speed accounting
- EZLOG™ for true 3-level reporting by account, project and user

Name Company ___ Address ____ _____ _____ State _____ Phone () SEND COUPON NOW FOR COMPLETE DETAILS ON ALL THE PACS BENEFITS

Signal Technology, Inc.

5951 Encina Road, Goleta, CA 93117 (805) 683-3771 **Outside California call toll-free (800) 235-5787** TWX 910-334-3471 VAX and DEC are trademarks of Digital Equipment Corporation

CIRCLE 214 ON READER CARD

PDP 11/70 RSX11M-PLUS TO VAX 11/780 VMS MIGRATION

By Allen Watson, Bergen Record, Hackensack, NJ

PRELUDE

"I wonder how you boot this baby?", I thought as I stood in front of our new VAX 11/780 late one Sunday night in August.

The hardware was almost ready, I knew — some remaining glitch in our terminal controllers. And the Friday before, Eric, our newly acquired programmer with VAX experience, told me that he had already brought up VMS for the first time. So it was bootable. But how?

Ironically, I was working the weekend because we were in the final stages of an RSX11M-PLUS Version 2.1 SYSGEN on our PDP 11/70. I was rebuilding all the task images that used the Files-11 libraries FCSRES and FCSFSL. So I was waiting around while the task builder ground away, with nothing to do until it finished, looking at my giant new toy, wanting to play.

"No panel switches," I mused. "Can't be that you enter an address... wait! Maybe from the console, like an 11/70 with a diagnostic panel." But the keyboard was unresponsive to normal typing.

Nothing was running, I knew that. I saw a familiar key in the power switch. So, I powered it off and on again. The keyboard was still dead.

Then, near the power switch, a rocker switch caught my eye. A label over it said: "BOOT."

I pursed my lips and frowned in doubt. "Nah! It can't be that easy." Actually said that out loud. But, like Alice with the cake labeled "Eat Me," a button labeled "BOOT" was too much temptation. I pushed it.

There was a whirring noise inside the blue box. I had a moment's panic at the unfamiliar noise, then I remembered: there is a floppy disk drive inside the thing. Something printed on the console! Something else, with a date in 1982, ("Have to set the date right," I thought to myself). Another line or two — pause — and VMS announced it was up, correcting itself by typing out the correct date and time.

I tapped "RETURN."

"Username:" said VMS. So I typed "SYSTEM" (a good guess, I thought).

"Password:" said VMS. "SYSTEM," I entered, since that is the username and password on a newly GEN'ed RSX system. But VMS declared it invalid. "Damn! Why'd they have to make it different? Oh, well, it was worth a try." I almost turned away, but then decided to give it one more crack. (I really had not read anything but one book on VMS concepts at this point.)

So I said "SYSTEM" again, and then "MANAGER" for the password — and VMS accepted me! Warm, fuzzy feeling. I made a mental note to change that login as soon as the system was live. If I could guess it in two tries, anyone could.

Forty-five minutes later I tore myself away from the console to check on my RSX task builds. And so began my love affair with the VAX.

WHY CHANGE FROM M-PLUS TO VAX/VMS?

It wasn't much more than a year ago that we migrated from RSX11M to RSX11M-PLUS. My article on that migration, previously published in The DEC* Professional (Volume 2, Number 3), was well received. And now here we are moving on again. Why?

The reasons this time were actually not as compelling as they were for the move from M to M-PLUS. We very nearly just added another 11/70 to our

stable, but when my company offered us the option of moving to VAX, we started looking into it. When we proposed the purchase of another 11/70, our upper management, who fortunately are familiar with computers, said, "How much more would it cost to move to VAX, and would it be worth it?" We told them it would cost quite a bit more, and their answer was basically, "You worry about getting what you need to do the job, and let us worry about paying the bills." It's a nice place to work.

Many More Users

A VAX 11/780 can support up to sixty-four interactive users with no messing around with non-standard equipment or software. I've heard of systems with more than that. We were about to add applications to our system that would increase the number of terminals to sixty-four (on two 11/70s) and we were worried about what would happen if one 11/70 went down. Response times on some interactive applications were already pretty bad, varying from five seconds when things were good to two or three minutes when things were bad. We figured that we needed at least two live 11/70s to handle sixtyfour users, and so wanted a third to spread things out and provide backup. But one VAX could handle them all (with a second for backup).

COBOL Programming, Limited Address Space, and Disk Access

Most of our applications were written (for reasons dictated by the company) in COBOL. When we started into the 11/70 world I pushed strongly for PASCAL or even C, because my investigations showed that COBOL on

RSX was terrible. At that time, COBOL-81 was just out and had very limited capabilities; I understand now it has improved and expanded. We used PDP-11 COBOL. And guess what? It was terrible.

When you use RMS files and FMS screen forms, those packages along with the COBOL run-time library take up 20K of your virtual address space, leaving you 12K to program in! Our COBOL tasks were insanely overlayed. We were unable to buffer our RMS files, so every read was a disk access, added on to all the overlay reads. I tested one program and found that for every record processed we averaged 100 disk accesses! (I never did quite figure that one out; there were actually seven or eight files involved with tables that had to be searched for each record, and so on.) Programs took six or seven hours to task build on a moderately busy system; compilations took many hours. Our programmers reported that on their best days they got three compilations a piece.

The latest versions of M and M-PLUS have begun to offer things such as User Data Space (an "extra" 32K for data structures), clustered libraries (but not for PDP-11 COBOL), and Supervisor mode libraries, all of which offer the programmer ways to get system code and data out of his limited, 32K virtual address space. But just six months ago most of it was "coming soon," and who knows when "soon" is with DEC? Any of these things would have helped, but would also have entailed rewriting a lot of code to take advantage of them. And we were figuring we would have to do it.

We also had high hopes in a "virtual COBOL" being developed by another vendor for RSX; I kept calling them, offering to be a test site, and they kept having delays. We never saw it, and as far as I know it still is not for sale. That, too, would have meant a lot of recoding and translating.

We had problems with file sharing, and we were using more and more memory as more and more users got on the system, each with a 32K program consisting mostly of RMS, FMS, and run-time COBOL stuff. I started looking into writing a generalized file

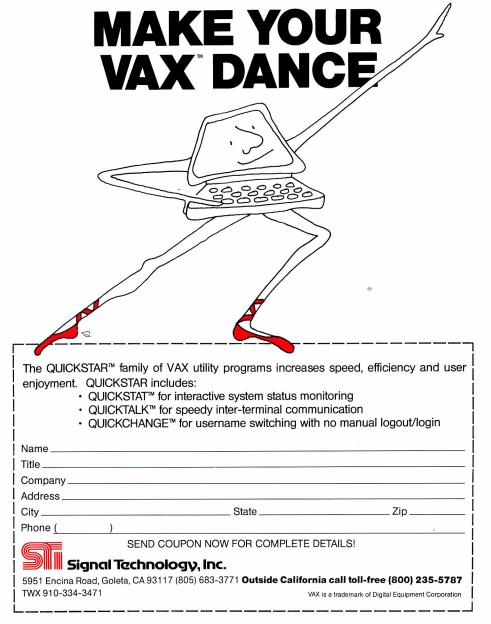
server for each of the most heavily accessed files. (There have been two very good presentations at the last two DECUS symposia on doing just that.) That way, we could remove the file-handling code from the application programs and have a single path to the file. Access conflicts could be resolved in that central file server.

So then we were told to look into the VAX. And suddenly all the problems magically cleared up.

The VAX is a virtual memory machine. This means that programmers don't have any address space limitation. Well, four gigabytes, but you call that a limit? We were busting a gut to try to get our programs to run. We knew all the shoehorn techniques, still had problems, and here was a machine that would take our programs and just say "OK, no sweat."

COBOL on the VAX is a dream. It has everything you could want in a COBOL compiler and more; you can almost write system programs in it. On top of that, it compiles fast and produces good machine code.

We did a benchmark. Took one of our medium-sized programs that only used two files, compiled and task built it on the 70, standalone, and ran it. I no



longer have the exact figures, but it was something like this: 28 minutes to compile, another 25 to task build, and one hour and thirty minutes to run.

We then talked to our local DEC office and got use of a VAX 11/780, also standalone. We took the programs and data down on a tape one morning, ran the programs through a conversion program and then spent about an hour finding and making five minor syntax changes the converter had missed. The files were copied over exactly as they were: RMS indexed files.

On the VAX, the compile took four minutes, the link took less than three seconds. No kidding! Our programmer hit "RETURN," turned away, looked back and saw a prompt, and figured he had to have done something wrong. He hadn't. And the program ran (using the same disk drives we have) in about eight minutes instead of one and one half hours.

For compile, build, and run, we were comparing 143 minutes on M-PLUS to under 13 minutes on the VAX, more than a ten-fold improvement. The run time was more than eleven times faster.

Now, we had timed some of our BIG programs — the monthly updates and such — at over thirty continuous hours (standalone) on the 11/70. If we had to run those when other users were on the system, forget it. If we could cut those thirty hour runs to three hours . . . if we could compile a program thirty times a day instead of three . . . We walked out of the DEC office fairly drooling with anticipation.

DEC does not claim the VAX is ten times faster than the 11/70. Maybe it's just the COBOL. Maybe it's a unique mix of things we were doing. But for us, if we had a chance of getting VAXs instead of 11/70s, we were going to go for it. We had reached the point of changing languages or changing machines, and the VAX clearly offered us an easy way out of our problems.

The 11/70 is a Dying Machine

Another reason for moving is that the 11/70 is no longer a front-line product with DEC; they have moved it into the Traditional Products group and have stopped making new machines. If we were going to buy another computer it seemed to be unwise to buy one that was clearly at the end of its life cycle. And they have not announced another PDP to replace it. They clearly wanted to encourage users to migrate to the VAX.

Rumors of the Death of RSX11M

At the DECUS Symposium in Anaheim during December, 1982, there was a session that was titled "Is there life after RSX?" Rumors were rife that DEC was going to stop development on the operating system and move on to other things. That was a little unnerving to say the least, and had no small influence on our decision to get out of RSX and into VMS.

Since that time, reality appears to be not as bad as anticipated. DEC has stopped adding new capability to RSX11M, although they continue to add support for new peripherals as they appear. They are, however, continuing to enhance RSX11M-PLUS. One reason is that the Professional 300 series of personal computers is based on M-PLUS, and as work is done on P/OS for the Professional there is a back-wash into the parent M-PLUS. Version 2.1 of M-PLUS recently added some very nice features, including (finally) a Supervisor Mode version of RMSRES that lets you get RMS out of your address space. RMSRES is also clusterable, and when (or if) the same thing is done for FMSRES, it will ameliorate some of the problems we were facing. But they are about one vear too late for us.

VMS, on the other hand, is a very live and growing system. New capabilities are continually being added, and dramatically useful ones are promised for next year (see the section on Cluster architecture that follows). It appears that DEC will be putting major effort into enhancing VMS for some time to come.

Frequency of Crashes

Ever since I have been a user of RSX, starting with RSX11M V3.1, crashes have been a problem. The pre-Symposium seminar in Crash Dump Analysis has continued to be in demand, so I presume I am not alone in this. We eliminated one cause after another, but, quite frankly we have

never arrived at an entirely reliable system.

Going from M to M-PLUS brought great improvement with the elimination of crashes due to exhausting Executive Pool Space. At first, we ran two weeks without a crash. But then we began to add users, and with increased file activity we began crashing two or three times a day. Examining the crash dumps showed a consistent failure with an odd-address trap and Pool Link errors.

From the Software Dispatch and other M-PLUS users I learned that the cause was a bug in IOSUB file termination routines. Some misplaced code was decrementing the outstanding I/O count after the file data structure had already been released back to free pool. When the system was busy, that word of memory was sometimes already reused by someone else and IOSUB was decrementing a byte of someone else's data structure, causing an odd address. The fix was published in the Software Dispatch and was included in Autopatch B for V2.0, and in V2.1. In addition there was another bug that could crash the system if you aborted PIP or SRD (and users frequently abort a DIRECTORY command after they see what they want).

The point is not the particular problem. System crashes seem to be part of living with RSX in any form, and with each new release there seem to be new reasons for crashing. From what I have been able to learn from users of VMS, however, crashes are extremely infrequent. Our experience so far would seem to bear that out, although we have not yet begun to load the VAX as heavily as we did the 11/70.

With many interactive users, with the daily distribution of our newspaper dependent on the computers, and with jobs that threatened to take days to run (with no possibility of checkpoint/restart), we simply could not live with a system that crashed as often as RSX. The VAX promised to cut down on both crashes and the length of time jobs had to run, making for a much more secure environment.

Difficulty of SYSGENs

An RSX SYSGEN has always been

an unpleasant task. Real old-timers probably think I am a crybaby; I understand the early versions of RSX took days just for the basic SYSGEN, whereas now it is down to three or four hours. But that is only for one basic system generation from start to finish. I have never done a SYSGEN that worked right the first time. Even with M-PLUS Version 2.0 we took about three weeks of night and weekend work to complete the gen.

We used non-standard hardware (foreign disks and controllers) and somehow DEC always managed to change the SAVE, BOOT, and disk driver routines so that our patches never worked on the next release; we always had to do it over. At one point I swore that DEC was doing it deliberately to make it tough on people who didn't use their hardware. On top of that, even our DEC hardware had CSRs and Vectors different from those recommended by DEC, so Autoconfigure, when it came along, would not work for us. And we could not change that because certain foreign hardware and software we use was locked into a particular set of CSRs and Vectors.

Because of earlier problems we had done a lot of customizing and optimizing on our system. And with each new SYSGEN, it got harder, not easier. We had to redo almost everything we had previously added to the system in order to maintain compatibility.

Furthermore, after you do a basic RSX SYSGEN, you still have the problem of rebuilding privileged tasks, rebuilding anything that uses resident libraries or Supervisor mode libraries changed by DEC, editing in task build options to DEC utilities, copying users files and directories onto the new system pack, and so on and so on. A new Autopatch tape or a new release, instead of being a welcome sight, was always greeted with "Oh, no! Not another SYSGEN!"

The VAX SYSGEN procedure by comparison is a piece of cake. Adding a device, for instance, only requires a reboot of the system because VMS reconfigures itself every time you boot. Our entire VMS system was set up in one day, including all the layered products. Of course we cheated a little: we

left the foreign hardware and software on the 11/70s so that everything on the VAX is DEC standard. We still bought non-DEC disk controllers and drives but we made sure they were guaranteed to be 100% compatible. Call that wisdom gained from experience.

We had three 11/70s; we bought two 11/780s, and are giving up only one of the 11/70s. So we will still have RSX around for quite awhile, but running only tried and tested software. M-PLUS V2.1 may be my last RSX SYSGEN, God willing.

Better Versions of Layered Products

We had Datatrieve, Forms Management System (FMS) and Record Management System (RMS) on our M-PLUS system. The VAX versions are vastly superior. When I saw them running on a VAX I almost didn't recognize them, especially Datatrieve (DTR). DTR on RSX, when the system is really loaded, is unbearably slow. Our experience with it was so discouraging that we really didn't use it at all, except as a tool for debugging and fixing up records that somehow got clobbered. I wrote a little DTR routine to handle an

BREAKTHROUGH IN VAX/VMS SESSION/IMAGE ACCOUNTING



Only TRACS™ allows examination and tracking of VMS session/image resource data by user, time interval, terminal port and other key fields. TRACS offers:

General query and interactive capabilities
 Full screen forms editor

Name ______

Title ____

Company ____

Address ____

City ______ State _____ Zip _____

SEND COUPON NOW FOR COMPLETE DETAILS!



Phone (

5951 Encina Road, Goleta, CA 93117 (805) 683-3771 **Outside California call toll-free (800) 235-5787** TWX 910-334-3471 VAX/VMS is a trademark of Digital Equipment Corporation



online phone list of about 200 records; even that was so slow no one used it, and I finally replaced it with a set of indirect command files that used FMS forms. Even Indirect was faster than DTR. Since getting our VAX, I again tried using DTR to do an online phone list, and it works like a champ. This time I used FMS forms from Datatrieve. I did the entire thing without reading the manuals, just using HELP, and did it in a day. And apart from my experience with FMS in my command-file phone list, I had never used FMS before.

The thing that makes VMS so pleasant is that everything works together. You can use FMS forms with DTR; you can call Datatrieve routines from any of the programming languages; you can write subroutines in one language and use them from any other language.

The new FMS and RMS utilities make working with those facilities simple. The ANALYZE utility can take an existing RMS file that was poorly designed and optimize it — automatically. We've already copied files from our RSX system over to VAX, optimized them, and put them back on RSX. Having the VAX has even helped us run our RSX system better, awaiting the day when we can transfer the programs directly onto the VAX.

One very nice thing about moving from M-PLUS to VMS is that so many things are familiar. EDT is identical. DTR does just what it does on M-PLUS plus much more; files and procedures can be copied over from RSX unchanged. FMS is, if not identical, similar. The forms created on RSX can be used on the VAX, although you may need to run an Upgrade program against them. DCL is very much the same, with many extensions. RMS files can simply be copied over (and then optimized). And if you want to, RSX tasks can be brought over and built and run in Compatibility Mode. More on that later.

Eventually, I suppose, many of the improvements in the layered products will make their way into M-PLUS — but not all of them. And clearly, the VAX gets the improvements first.

VAXCluster Announcement

The clincher for us was the an-

nouncement during the St. Louis Symposium of VAXClusters. We had already ordered our VAXs, but when we heard that announcement we knew we had made the right choice. The cluster hardware (HSC-50 intelligent disk controllers, the Computer Interconnect or CI bus, and the Star Coupler) is already available and supported on the 780; VMS V3.4 will support it on the 750s. And VMS V4.0, due out next spring, will implement the full-blown cluster concept along with many new capabilities including checkpoint/restart of processes, full sharing of disk storage between CPUs, cluster-wide record locking, common journaling to track and safeguard file activity, and the ability to roll your files forward or backward by defined recovery units.

These capabilities offer us unprecedented opportunity for smooth growth. It opens the door for us to look at adding an online library system for our newpaper "morgue" with full text searching for everything our paper has published, an online circulation system, company-wide word processing, and ultimately, full computer-to-plate newspaper pagination including graphics and photos. With the cluster architecture we can add terminals, memory, disks, and additional VAX processors and still have a single, integrated system.

LOSSES AND PROBLEMS

Not everything about the move to the VAX is a gain. There were some areas in which we lost work we had done on M-PLUS, and some where we had to fight to retain M-PLUS features.

Shadow Disk

Readers of my previous article on "M TO M-PLUS MIGRATION" will remember that the ability to "shadow" disk volumes was one of our major reasons for going to M-PLUS. Shadowing provides an automatic, constant, on-line, mirror-image copy of a disk on a second drive. Everything that happens on your primary disk is instantly mirrored on the shadow disk. If the primary drive fails you can immediately resume operations from the shadow pack.

VMS does not have shadowing as part of its repertoire. That was a major

stumbling block to us. The new cluster architecture will (sometime in 1984) provide shadowing on the VAX for all RA series disks, whether they are attached to the HSC-50 or a Unibus controller. The HSC-50 will actually do its own shadowing, removing that burden from the host VAX. But that shadowing ability is not currently present in VMS, and when it comes it will not handle Massbus disk drives such as the ten we now have.

The reason DEC gives for this is that the RA series disks appear to the system to be error-free — no bad blocks — because the controller handles remapping of bad blocks. Massbus controllers don't, so the system itself must remap bad blocks. We hope that there will be a way to enable shadowing for Massbus disks that are certified error-free.

To use the new VMS shadowing it appeared that we would have to junk all our CDC drives and buy new DEC controllers and disks. And the software wouldn't be ready until next year. We wanted the VAX; we were not willing to give up shadowing because we need that instant backup; and there was no way we were going to get rid of perfectly good disk drives. So we pressed DEC, making shadowing of our existing disks a condition of the sale. And DEC came through.

It seems that in Munich, Germany, someone else had felt as we did. The DEC software people in Germany had written a shadowing package for VMS that operates identically to the M-PLUS shadowing. It cost us extra, of course, but it is DEC supported software. I believe that The Record is the first VAX site in the USA with disk shadowing for Massbus disks. As on M-PLUS, all of the packs in a shadow set must be set up with identical bad block files; if a block is bad on one pack it must be declared bad on all packs in the set. This is how M-PLUS gets around the bad block problem. (Readers interested in the VMS Shadow package should contact the DEC Sales Office in Piscataway, New Jersey.)

I am not certain what the future of this package will be in forthcoming releases of VMS. I suspect we may run into conflicts between this method of shadowing and the new one. Come

what may, we intend to continue shadowing all of our vital disks, of whatever kind.

Command Files Are Different

On the VAX there is no MCR unless you run in compatibility mode, which is slower than running on an 11/70. Indirect Command Files created on RSX must be entirely rewritten in VAX DCL. When you have hundreds of command files as we do, that is no small problem. It isn't insurmountable; Command Procedures on VAX can do almost everything Indirect Command Files can on RSX, and in fact can do many things you cannot do on RSX. But it is a major hassle you must be prepared for if you make this migration.

Alternate Command Line Interpreters

There seems to be some provision in VMS for alternate Command Line Interpreters (CLIs), but so far your only options are DCL or MCR (compatibility mode). I haven't been able to find how to create your own user CLI, which is a well-documented feature of M-PLUS since Version 2.0. We particularly miss CCL, which was a CLI obtained through the DECUS free software tapes. We had altered CCL slightly to create two alternate CLIs for two of our application packages, restricting them to commands related to their application. We are still struggling with how to implement this on the VAX. Many of the pieces seem to be there but the big picture hasn't emerged yet.

Among the "pieces" I refer to are the ability to define foreign commands, and even to define new DCL commands, without any compilation of programs. This gives us a large part of the capabilities of CCL that we liked so much. For example, simply by typing: H:= = SET DEFAULT SYS\$LOGIN a user can define the command "H" to set him back to his home directory. very useful on VMS where directory wandering is a way of life. If he includes that line in his LOGIN command file it will be defined for him every time he logs on. Other users can define "H" to mean something entirely different.

New DCL commands can be added for individuals or to the system by using the SET COMMAND command. So



with software from McHugh, Freeman and Associates, Inc.

- **1. Data Encryption:** Encrypts data, source and task image files
 - Prevents disclosure of sensitive data
 - Isolates sensitive programs (from privileged users)
 - Prevents unauthorized modifications

2. Menu/Access Authorization:

Allows only menu access to all applications programs

- Limits display to only those items available to user
- Prevents access to the monitor level
- Traces user movements through the menu system
- **3. Keyboard Monitoring:** Monitors and interacts with any or all terminals on the system
 - Use at your discretion or as a function of the log-in sequence
 - Optionally creates a log of all user activity
- **4. Password Changing:** Changes passwords routinely and quickly
 - Choose from a list of over 3,000 encrypted English words or alphanumeric sequences

Contact us for full product specifications and demonstration packages.



McHugh, Freeman and Associates, Inc. 1135 Legion Drive Elm Grove, WI 53122 (414) 784-8250

CIRCLE 57 ON READER CARD

far we have very limited experience with this, but it seems to offer a path to what we want for user-specific CLIs. At least we have the ability to add commands. Deleting or restricting commands already defined for all users seems a bit more

seems a bit more difficult.

However, any command that has remote the possibility damaging something is protected by VMS "privileges," so perhaps we are worrying needlessly. Restricting user privileges is a built-in part of VMS, and is far more selective than on RSX. RSX has two classes of users, privileged and non-privileged. On VMS there are thirty-one different privileges that can selectively assigned to users by their login. For example, privilege GRPNAM allows the user who has it to insert logical group names in the name table. But it takes the SYSNAM privilege to create system-wide logical names. Part of what we were accomplishing with alternate CLIs on RSX can therefore be controlled through careful assignment of privileges.

Special Device Drivers and Privileged Tasks

We have one special device driver on our M-PLUS system for an analogto-digital converter. Device drivers must be rewritten for VMS because the operating system handles interrupts differently than RSX. I have no reading on the difficulty of this task because we haven't started to convert that application yet. Users who have many special drivers may want to investigate DECnet I/O calls will also not run, but must be rewritten in VAX code. And applications that involve cooperating tasks may require some modification.

Fortunately for us, we had very few privileged tasks other than those

provided by DEC; I would say about a dozen programs have to be redone. The restriction on RMS-11 file sharing, however, hit us hard. One major application with over 120 programs does extensive file sharing. We cannot use this in compatibility mode, and so must wait until we convert it entirely to native mode before bringing up the application on the VAX.

In my opinion, no one should consider running permanently in compatibility mode. It's like buying a Cadillac and installing a VW engine from vour old beetle. The new car goes slower than the old one. So perhaps the restrictions on compatibility mode are a hidden blessing, forcing us into conversion native mode from the start.

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 POWERFUL.

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.

FINAR IS EASY TO LEARN.

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.

FINAR IS EASY TO USE.

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) 664-1172.



JAMES B. HOTZE & CO.

6101 Southwest Freeway, Suite 406 Houston, Texas 77057 (713) 664-1172

CIRCLE 51 ON READER CARD

this carefully before deciding to undertake a conversion to VAX.

Furthermore, privileged tasks from RSX will not run on the VAX, even in compatibility mode. They have to be rewritten. And non-privileged tasks that use memory management directives, RMS-11 file sharing, or

FMS Differences

The FMS on RSX differs from VAX FMS. We have

found that every line of source code that calls FMS routines has to be edited into a different format. This will probably double the expected conversion time for the programs involving FMS.

Here's an example one of my COBOL programmers gave me:

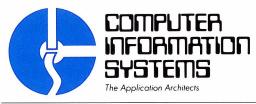
VAX/VMS Resource Accounting Software

A dozen fresh reasons to inspect CIS' package



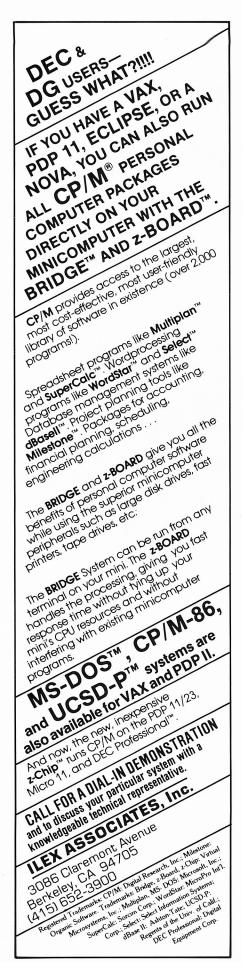
- 1. Image level accounting
- 2. Project level accounting
- 3. Processes a VAX network
- 4. Integrated security system
- 5. Incorruptible file structure
- 6. Wildcarding for selections
- 7. User definable rate structure
- 8. Budgeting by user and project
- 9. Program callable interface
- 10. Menu driven with on-line help
- 11. Toll-free telephone support
- 12. CIS Applications Architecture

CIS. When it comes to VMS Resource Accounting and Chargeback Systems, we put all the right eggs in one package.



165 Bay State Drive / Braintree, MA 02184

617-848-7515 / Outside of Mass. call toll free: 800-232-5215



CIRCLE 219 ON READER CARD

RSX VERSION:

CALL "FPUT" USING D-SUMMRY-CHANGE, TERMINATOR, N-SUMMRY-CHANGE VAX VERSION:

CALL "FDV\$PUT" USING BY DESCRIPTOR D-SUMMRY-CHANGE, BY REFERENCE STAT, BY DESCRIPTOR N-SUMMRY-CHANGE

The VAX version shown is for Version 1 of VAX FMS; we have Version 2.0 installed but have not yet received anything except the release notes. They indicate that V2.0 will support most V1 FMS calls, but such support is not guaranteed in later releases, so additional editing will be required.

Sheer Volume of Documentation

I suppose any change of operating systems runs up against the problem of absorbing all the new documentation, but on VAX it seems especially formidable. The basic DOCSET is sixteen three-inch looseleaf binders. There are several more for FMS, DECnet, the Common Data Dictionary, and Datatrieve. My guess at the number of pages to be absorbed is somewhere around 13,000.

The DEC classes we've had so far have been outstanding. I would advise you, however, to budget a large amount for education the year you buy your VAX. Each week-long course costs about \$1100 per person. We are training eighteen programmers and twelve operators, and have budgeted close to \$100,000 for training. Not everyone is going to all the courses, either; we are sending a select few and having them hold in-house training for the rest.

CONCLUSIONS

We have purchased our VAXs, so obviously we felt the gains outweighed the losses and problems. We have not yet begun heavy use of the VAX; primarily, for now, the VAX is being used for training and conversion efforts. We have no one but programmers using it. The proof of the pudding will come later. I intend to write additional articles giving more of our ex-

perience with the VAX, comparing performance and capacity of the 11/70 with M-PLUS and VAX/VMS.

Users with many real-time applications, most of which involve foreign device drivers, should consider carefully before abandoning their PDP-11s. The VAX does not have a reputation as a real-time machine, and the conversion effort would be considerable. The gains may make it worth your while, but you will have to judge that for yourself.

Users with business-type applications, particularly those using COBOL, ought to think seriously about getting a VAX. The performance difference in COBOL is nothing short of incredible.

And if you write a lot of large applications that are continually bumping up against the virtual address limit on RSX — well, that's what a Virtual Memory System (VMS) is all about. It removes the limit. You need a VAX.

If you think you may be interested in a VAX I would advise you to read two DEC documents, obtainable from your DEC sales office:

VAX TECHNICAL SUMMARY (EJ-21977) VAXCLUSTER TECHNICAL SUMMARY (booklet)

The first is a magazine size book that summarizes all of the hardware and software on a VAX system, including the major layered products and languages. I found it invaluable in knowing what to expect from a VAX and what software products to purchase with the system. Simply put, after I read this, I knew I wanted one.

The cluster summary is a small booklet that adds a description of the current and planned features of the VAXCluster Architecture. From this you will learn where VAX systems are going in the near future so as to plan how you can use them over the next several years.

Get a reductive repert can six delivering an word performance from your shair memo. DEC Compute

Now available

PRO/RT-11.

Contel software can improve the performance of your DEC system while reducing your programming costs. Our repertoire of proven software products can simplify database management, deliver multi-user, distributed processing and networking capabilities, add word processing to every terminal,

even expand operating system capabilities. For example:

RTFILE®: Interactive, relational database management system for menu-driven, forms-oriented data processing without programming.

SHARE-eleven: High throughput, memory-resident timesharing system

which supports up to four users.

hensive, disk-swapping timesharing system which can support as many as 20 local and/or remote users.

STAR-eleven: High-performance local area networking system which links multiple PDP-II and/or LSI-II computers for resource sharing and distributed database processing.

LEX-11: Powerful word processing system with more functionality than the standalones and the flexibility of the DEC environment.

DISKS-eleven: Field-proven sub-device utility which divides large disks into any number of smaller virtual disks.

These and Contel's other performance packages can enhance your DEC computer. Get the complete listing by contacting Contel, the big name in small systems software. Write Loretta Kane, Contel Information Systems, 4330 East-West Highway, Bethesda, MD 20814, or call 301/654-9120.

CONTEL

RTFILE registered T.M. of Contel Information Systems *T.M. S&H Computer Systems, Inc.

MicroVAX 1 and VAXElan: A Realtime Marriage

By Robert B. Robbins, Array Computer Consultants, Inc.

Execution of real time applications tends to differ from non-real time applications in that they are usually event or interrupt-devices and time critical. These real time characteristics force a desire for full control of the central processing unit (CPU) and, therefore, need to be dedicated single user applications. VMS is a multi-user operating system, and has all of the security and user accounting routines necessary to control the movements of all of its users built into the system. In order to run on VMS, real time users must sacrifice some speed due to these built in routines.

Recently applications designers had two new options added to the varying choices of hardware and software available for product development. Digital Equipment Corporation has announced a realtime programming toolkit called VAXElan which offers the ability to develop dedicated, realtime systems for VAX processors. In addition, the long awaited MicroVAX 1, a super microcomputer, running the MicroVMS operating system, has been introduced, offering VAX architecture and performance at microcomputer size and price.

The Configuration

The VAXElan development tools run on both the VAX/VMS and MicroVMS operating systems. Applications can be developed on any of the VAX processors from the top of the line 11/780 or 11/782 to the new streamlined 11/725. The VAXElan toolkit requires approximately 2000 blocks of disk storage to support the entire VAXElan environment. This includes the optimized EPASCAL compiler, runtime system, program builder, system builder, the EDEBUG symbolic PASCAL debugger, HELP files, and INCLUDE libraries.

The VAXElan target system, the executing dedicated application, will

run on VAXs from the 11/750 processor down to the low end 11/725 and MicroVAX 1. VAXElan application systems normally require target processors with a minimum of 256 KBytes of memory, but will support memory up to the physical maximum of a given machine.

The MicroVAX 1 offers a compact unit with the VAX 32-bit architecture. The CPU, Winchester and floppy disk storage, I/O and communications ports (Q-BUS) and power supply come in a cabinet which is small enough to fit under a desk (the height of a two drawer file cabinet). It maintains complete VAX compatibility with the exception of the PDP-11 compatibility mode. The MicroVAX 1 supports four gigabytes of virtual address space, 16 32-bit general purpose registers and all native mode instructions.

The key to VAXElan superiority is that it offers the power of the 32-bit VAX architecture and the sophistication of the Elan PASCAL language. With the availability of both UNIBUS and Q-BUS VAX processors, both types supported by VAXElan, the performance, flexibility and mobility across the VAX family of processors is maintained.

The Environment

The VAXEIan development toolbox offers a full application environment. The optimized VAXElan PASCAL compiler has been found to significantly reduce the size and improve the speed of application programs. It takes advantage of and uses the VAX general purpose registers and the hardware stack capabilities. EPASCAL is a block oriented compiler, allowing program modules to be written and compiled separately, without sacrificing the PASCAL typechecking features between modules. Because it compiles with the International Standards Organization (ISO) specifications, source code from other compilers is totally compatible.

The modularity of VAXEIan PASCAL allows for the support of global data. This offers inter-process communication, much more difficult if not impossible in most of the other five (5) DEC PASCAL-like compilers. With the extensive and sophisticated syntax support for realtime needs, VAXEIan PASCAL allows for interrupt dispatch and control, event response and synchronization, semaphore and exception condition handling, device driver development and implementation, and process scheduling, creation, deletion and control, just to name a few.

In addition, EPASCAL includes numerous procedures and functions which provided many useful services. In many cases these procedures and functions parallel those system services which would otherwise be provided for by VAX/VMS. These include device allocation, deletion and control, memory allocation, deletion and control, message creation and manipulation, extensive bit and string checking and manipulation (a welcome addition), UNIBUS and Q-BUS control, and logic checking.

It should be obvious from this overview that the developers, though they didn't grant everything on our realtime wish-list, made some major advances toward a true realtime, high level language. In my own use of the compiler over the past two months I have found it to be, without a doubt, one of the most sophisticated on the market.

Building An Application System

As program modules are written, they can be separately compiled and placed in libraries for later use, when needed for compilation of other modules offers the ultimate in structured design and implementation.



Program linking is handled by the VAX/VMS LINK utility. The VMS LINKer is useable because the EPASCAL compiler produces VAX-11 object code. This enables the user, under certain circumstances, to link other VMS object code with EPASCAL generated object code. The result of the linking is executable code (i.e., "FILENAME.EXE"). This executable code is not normally runnable under VMS. It is the input file to the VAXEIan System Build Utility (EBUILD).

The EBUILD Utility molds a tailored application out of many components. It combines the user programs and device drivers with the necessary routines selected from the VAXElan runtime library of modules such that only the needed functions are present. It then adds in the VAXElan Kernel to complete the system.

This complete system (the executable application) is now transferable to the target processor, or can be booted

to run on the development processor. The systems can be transferred by one of two methods: either the application is downline loaded over Ethernet; or written to disk storage medium for bootstrapping.

Once the application system is bootable it can be debugged using the VAXElan symbolic PASCAL debugger, EDEBUG (pronounced "Eat a bug"). This symbolic debugger is similar to the VMS DEBUG. The Elan debugger can control the execution of the application either from the development system, over Ethernet, or stand-alone within the execution system. In addition, the debugger offers realtime process control features not previously available under VMS. With EDEBUG the user can not only do full program debugging in PASCAL, but EDEBUG also enables the user to view the application's concurrently executing processes on one terminal, while displaying the mainline PASCAL program via another terminal. EDEBUG also contains process and job performance monitoring capabilities.

An important fact to realize is that the executing application has no operating system. When the processor is booted the application begins to run and continues until completion. The only control over the executing system is that which is built in to the application via menus, command language, etc. Of course, the more sophisticated the control, the more cumbersome and less efficient the performance of the application.

Single User, Multitasking

VAXElan produces a single user, multitasking environment. That is, there is only one user running jobs (programs) within the system. The multiuser capabilities of VMS are stripped out including all of the realtime headaches. There is no swapping, or paging as the application is totally memory resident. The jobs which are running can create any number of processes (similar to sub-processes in VMS) which will run in parallel, based on the design of the specific application. The processes each have a run priority which determines the hierarchy of execution. In addition, the scheduler allows CPU control based on a combination of the aforementioned priorities and on the status of each

The smoothest path between RSTS/E and VAX / VMS just got smoother: there's a major new release of

ROSS/V

ROSS/V has always provided:

- the fastest way to bring up RSTS/E applications on the VAX.
- the only way to do RSTS/E development on the VAX.
- an extensive subset of RSTS/E monitor calls and standard RSTS/E features, like CCLs, DOS-formatted magtape, and RSTS/E-style file update mode.

Now, in Version 3, ROSS/V supports:

- the "hidden" RSX run-time system (with 32 KW job size).
- · resident libraries.
- job spawning and detached jobs.
- spooling to VMS print and batch queues.
- mailbox send/receive for communication with VAX-11 BASIC and other native mode applications.

How ROSS/V works:

ROSS/V is written in VAX-11 MACRO, and RSTS/E monitor calls are performed in VAX native mode. The rest of your PDP-11 code (in applications, run-time systems, TKB, etc.) is executed directly in the PDP-11 microcode that's present in every VAX. ROSS/V runs under VMS, not in place of it. Thus, some users may be working under the RSTS/E subsystem provided by ROSS/V while others are concurrently using any of the other VAX/VMS capabilities.

Call or write for the new ROSS/V technical summary, which describes all of ROSS/V's features.

Evans Griffiths & Hart, Inc.

55 Waltham Street Lexington, MA 02173 (617) 861-0670

OnLine Data Processing, Inc.

N. 637 Hamilton Spokane, WA 99202 (509) 484-3400

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

CIRCLE 176 ON READER CARD

ROSS/V

process. A process can exist in one of four states. The SUSPEND state indicates that the job or process has been placed in a state of temporary pause. A WAIT state indicates that the process is not capable of executing at that instance due to any number of reasons (waiting for an event flag or interrupt, lack of resources, etc.). Any number of processes can be in the READY state, that is ready to execute at a given time. but lacking the priority to execute. Only one process can be in the RUN state at a given instant. The RUNning process has control of the CPU and is the process with the highest priority and is in an executable state. The above described competition of processes running in parallel allows for extremely efficient realtime applications.

Performance

The jury is still out on VAXElan performance. There has been some benchmark testing done comparing VAXElan applications run on the MicroVAX 1 compared to applications run under VMS on the 11/730 and 11/750. The VAXElan performance on the MicroVAX 1 is better than the 11/730 but not quite as good as the 11/750. In the case of interrupt processing preliminary tests indicate that response to interrupts to the point of user mode processings in VAXElan is considerably better. This maybe due to the more efficient way in which VAXElan device drivers are written. Caution should be taken when using these figures as the benchmarks are very preliminary and were done for specific instruction types.

Conclusion

The world of VAX realtime has, in the past, had to turn the VMS operating system into a single-user operating system, and suffer with the multi-user hooks and accounting problems which are embedded therein. Realtime designers have had to justify a large 32-bit processor when a smaller one would have been sufficient. The marriage of VAXElan and MicroVAX 1 solves these two and many other realtime problems while, at the same time, gaining features which, in the future, will make realtime, distributed processing a less ominous design area.

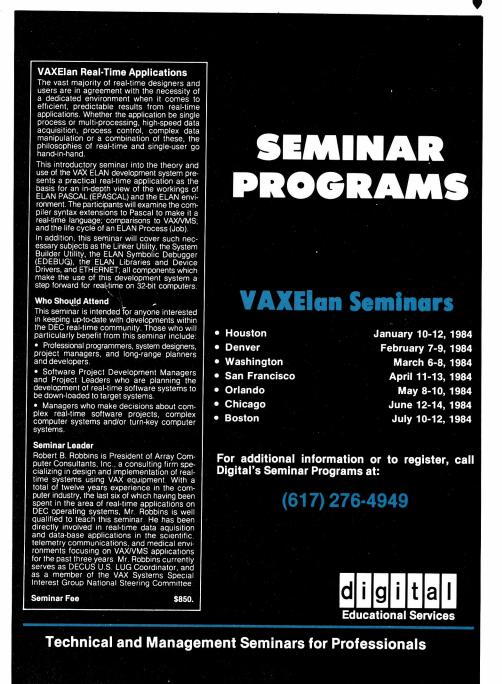
The user shouldn't be lulled into

thinking that with the additon of VAXEIan and MicroVAX 1, VAX realtime will be able to compete test for test against the likes of the SEL or high-speed HP processors. VAXEIan has not solved the VAX MIPS (Millions of Instructions Per Second) problem. It has only made the obvious superiority of the VAX as an all-around processor more justifiable.

This overview of VAXEIan and MicroVAX 1 is meant to be an introduction. It does not touch on the beauty and sophistication of the EPASCAL compiler syntax, but at-

tempts to give the reader a taste for what this new realtime toolkit can provide to applications and OEM system designers. I have not attempted here to discuss the clean and efficient way in which VAXElan deals with Ethernet communications; it deserves an article to itself.

I hope that this article gives the reader enough information to delve intelligently into the possible uses of the VAX for realtime processing both through the VMS operating system and now though the use of the VAXElan realtime development toolkit.



THE PRICE OF SLOTH/EDT

By Ilan Caron, Kibbutz Ga'ash, Israel

There has to be a moral to this tale.

This is the story of how I faked myself out using EDT.

As every school kid knows, EDT has a great RECOVER mechanism. So, naturally, when the terminal I was using (a VISUAL 100, which I will discuss later) started freaking out, I said to myself "What, me worry?" I was in the middle of an EDT session (two-hours long). I kept my cool even when my terminal started auto-repeating the last key pressed about 132 (strangely enough) times. In this case, the last key pressed was the CTRL/U combination. Wonderful . . . all of a sudden EDT started devouring up and generally wiping out the invaluable RNO file I was working on.

OK, first things first. I managed to CTRL/C my EDT job out of existence (my VISUAL 100 kindly broadcast about 132 CTRL/Cs, just to make sure). At this point, I had a JOURNAL file on disk, containing a record of the entire session. Including all the garbage that resulted from my (psychedelic) VISUAL 100 freak-out.

It was at this point that I decided to be systematic (which proved to be my downfall). I even wrote down an ad hoc algorithm to recover my EDT session.

- 1. Disable the DZ line I was working on, (the VISUAL 100 was still transmitting endless SOS messages . . .)
- 2. Move to another terminal, this time choosing something reliable like a VT100 (good propaganda for DEC).
 - 3. Backup the journal file, just in case . . .
 - 4. Backup the RNO file, just in case . . .
- 5. Edit the journal file to wipe out all the (VISUAL 100) garbage.
 - 6. Rerun EDT, with the RECOVER option.
 - 7. Continue working as if nothing had happened.

That was the theory, now for the implementation. Steps (1) and (2) went fine.

step (3): PIP X.JOU = FILE.JOU

step (4): PIP X.RNO = FILE.RNO

step (5): EDT X.JOU

Here I wiped out all the garbage at the end of the journal file, using the backup version, just in case . . .

step (6): EDT FILE.RNO/RECOVER

• EDT recovered everything, just as it says in the book, including all the garbage, because I hadn't edited the original journal file. I obviously didn't want that, so I KILLed myself

(UTILTY), thereby saving the journal file. If I had QUIT, the journal file would have vanished into thin air. I don't use the QUIT/SAVE option; it clutters up directories. Anyway, the VT100 I was using was doing a good imitation of a freaked-out VISUAL, so I couldn't very well QUIT.

Next I tried: EDT FILE.RNO/RECOVER:X. This was intended to RECOVER my RNO file, but now using the edited and correct version of the journal file. I naturally got a SYNTAX ERROR from EDT. My conclusion: you can't specify a recover file whose name is different from the input file. At least not like that. That's where my basic laziness reared its ugly head. I admit it; I couldn't be bothered to go to the shelves for the (very good) EDT reference manual.

At that point, I said to myself "what the hell, let's shortcut!" This is something I'm famous for advising people to avoid (at all costs). So I proceeded to shortcut:

1. PIP FILE.JOU/DE

(I didn't need it anymore; I mean I had the backup version, didn't I?)

2. PIP FILE.JOU = X.JOU/RE

(Renaming the corrected journal file, so I could EDT FILE.RNO/RECOVER with no problems.)

And what happened? You guessed it . . .

?Can't find file or account — X.JOU — continuing

Why? Because when I edited the backup version of the journal file (X.JOU), EDT did a bit of self-mutilation. When I EXITed normally from EDT, EDT kindly zapped all journal files of the name X.JOU (because that was the filename prefix of the file I was editing plus the standard journal extension and who needs all that garbage if the EDT session terminates successfully anyway?) . . . which left me high and dry.

I did manage to somehow scrounge around among the TEMP??.TMP files and in that way recover my RNO file. But at this point my tail was certainly between my legs. And it was at the last moment. I had almost logged out, with the intention of entering a nature retreat, and maybe learning a new profession, like sky-diving; something safe, where I couldn't inflict any more damage on myself.

The moral: THINK (as they say in IBMland).

DEAR VAX/ RSTS MAN



Send questions to: DEAR VAX/RSTS MAN P.O. Box 361 Ft. Washington, PA 19034-0361

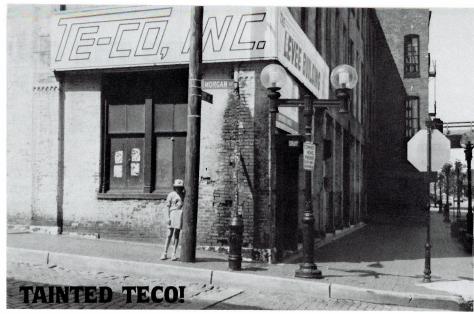
Dear VAX/RSTS Man:

Although our Du Pont site at Corunna will be installing Version 2 of BASIC +2 shortly, we have encountered consistent problems with BASIC +2 V1.6 programs, task built and run, under V7.2 of RSTS. If we open an RMS file in our BASIC +2 program, whether by literal or variable name with a protection code (using the format: "... <40>" or ".../PROT:40" for example), the RMS error #1824 occurs (meaning: explicit or default file spec contains extraneous field). Is there a way to alleviate this annoying problem?

Terry G. Ridgers, Systems Analyst DuPont, Canada Inc.

Dear Terry:

The standard RSTS method of specifying file protection codes (for example < 60>) has never been supported by RMS. The BASIC +2 product under RSTS supports both "native" files and RMS files, but you can specify a protection code within the file name for native files only. For files opened by BASIC without special RMS clauses, specifying RSTS protection codes within the file should work. If it doesn't, check to be sure that you built BASIC under RSTS with "native" as the default file type.



SUBMITTED BY DAN STOLLER, DATA MANAGEMENT ASSOCIATES, INC., ROCKVILLE, CT

Smooth transition — the activity required to change — is often as crucial as the change itself. **DISC** recognizes how pivotal the process of evolution can be in business computing applications. **DBL**, the business language for DEC PDP-11 and VAX computers, reflects this concern in one of its major design considerations: **Portability**.

DBL portability means a broad capacity to transport your existing software as you outgrow your hardware. It means ease of installation, swift modification of files and a concentration on development — not maintenance. The cost and complexity of the upgrading procedure is greatly diminished by virtue of DBL's inherent upward compatibility.

Smooth passage in an environment of perpetual change: The DISC commitment to the DEC user . . . through DBL.



3336 BRADSHAW ROAD SUITE 340 SACRAMENTO, CA 95827 916/363-7385 TWX 910/367-3701

OFFICES IN AUSTRALIA AND ENGLAND

THE VAX DRIVES THE COUNTRY'S 1ST FULL CHANNEL TELETEXT SERVICE

By Marci Shatzman, Time Video Information Services Photos By Nancy Kessler

The staff had been hired. The offices had been set aside on the 5th floor of the Time and Life Building in Rockefeller Center. The potted plants were even in place. But in the summer of '81, a couple of strange-looking machines called Norpak IPSs and a few standard video display terminals, Digital VT102s to be exact, were all there was to Time Incorporated's teletext service.

Time had been an innovator in cable television. It had gotten into the business early and had stayed with its fledgling Home Box Office (HBO) for years before it became profitable. The company also owns American T.V. & Communications Corporpation, the largest cable franchise in the country. Given that history, Time was determined to become a pioneer in teletext, one of two cable TV capabilities being billed as the new mass information mediums of the 1990s.

Teletext and its counterpart, videotex, are methods of sending computer-generated information and graphics into the home. Both allow viewers to sit in front of their TV sets, turn to a cable TV channel, and using a remote control keypad or keyboard, call up a variety of information at will from news to stocks to what restaurant to pick for dinner that night.

Teletext has its limitations. It's only one-way. So it can't be used for messaging, banking or shopping by computer. Videotex, which is two-way, can. But there's very little two-way cable in the country. And Time opted for teletext to keep subscriber cost low and to get into the marketplace quickly.

Only one obstacle remained. No one had done it before. Time wanted to do full channel teletext. That meant managing a real-time database of several thousand screens of graphics and text. "That kind of thing, especially us-

ing NAPLPS graphics and producing a teletext video signal on a full video channel, had never been done before," said David McCune, former general editor for TVIS (Time Video Information Services) and now president of The Proteus Group, Inc., a software development and consulting company for the videotex/teletext industry. (North American Presentation Level Protocol Syntax is the name for a standard of transmitting graphic information over any medium.)

Time looked to two Canadian companies for help in developing its new brainchild. For the hardware it went to the Ottawa firm called Norpak.

For the software it chose Infomart, which already was operating a commercial videotex system for farmers in western Canada (Grassroots) and has installed 17 teletext and video systems worldwide. It also had an agreement with the Times Mirror Co. of Los Angeles, California on its videotex project.

Infomart put out bids for help in writing the response to Time, Inc. Enter Co-Triple Computer Systems, Inc., of Markham, Canada, which was to play an integral role.

Once established, the software development team was headed by Chris Howlett, president of Co-Triple and Doug Chapman, manager of Infomart. At teletext the mission fell to John Lopinto, the technical chief and teletext editors, David McCune and David Spohn.

Chapman said a given at the outset was the use of a Vax 750.

"The VAX provided a small and medium family that were affordable. They also had good communications capabilities. If you were to compare them with, say an IBM system, they were cheap and easy to interface," he said.

"We didn't even look at anything

else. Infomart used VAXs in its services already. We hoped initially to use the software."

The VAX software Informart used is called ITSS. A slash T was added to the teletext software to differentiate it from the original. But as Chapman points out, "they're entirely different."

"The primary function of the 750 was to manage a database," McCune said. "We always knew there would be a need for a lot of data storage to be transferred back and forth to whatever would insert the data into a video signal. That machine didn't exist yet."

"The question for us was, what do we use to create this data, a VAX or some other machine."

The Time, Inc. development team looked at several alternatives to this initial dilemna. They investigated Exxon's word processing system, Wand, the DEC TMS-11 and DEC's typesetting and the word processing system for the VAX.

"We talked to DEC about taking out pieces of already existing VAX software," McCune recalled. "We had DEC people in here and we explained to them what we do and what our needs were. They were baffled by the whole idea. It was outside of their realm of experience so we didn't get very far with that."

Lopinto and McCune checked out the Radio Shack TRS80 PC with an eye toward using it as a front-end computer on which to edit the text before transferring it into the VAX.

It was one of two methods that had to be decided on at that early juncture.

"We toyed with these basic approaches — having the editorial interface go via some front-end processor like a micro, or having the VAX handle the entire editorial input," McCune said.

"There were advantages and disadvantages to each," he said. "The

VAX is a fast mini-computer. It's a very fast number cruncher if you're doing things like mathematics. But if you just hook a VT100 up to a VAX, it handles I/O relatively inefficiently. That's not what it's made for."

The problem was that the word processing programs "that already existed on any micro we looked at couldn't format the pages the way the editorial people wanted them without modifications to the source code," McCune said. And "the leap to systems with layout capabilities was overkill. There are a few, but 98 percent of what those costly systems did we didn't want."

McCune says, "We were looking for a quick fix, something off the shelf. We wanted to concentrate the software development on two areas — database management and teletext transmission management. The principle was that the editorial staff could always get by. If the other two areas didn't work, it didn't make any difference how good the editorial product was."

The final choice: EDT, the text editor which comes with the VMS operating system.

"EDT seemed to a lot of people to be a very bizarre choice," McCune said. "It's far less sophisticated a word processing system than most anything else we looked at. But you could make it do all the formatting that anyone had asked for."

Those absolutes were to be able to create on the screen an approximation of what the actual page would look like. McCune calls it a what-you-see-is-what-you-get-approach.

"What we didn't want was a system where formatting and composition commands were imbedded in the text like they are in Digital Standard Runoff, the system DEC gives you free with the VAX. That has to be the most unwieldy formatting system ever created by man.

McCune describes the VAX's beginnings at teletext as "unceremonious." "It didn't get no respect..." he laughed, quoting comedian Rodney Dangerfield.

The 750 was plugged in on the fourth floor of Time, Inc.'s computer center in the Meadowlands (Lyndhurst, New Jersey). "The whole floor was fill-

ed with big computers (mostly IBMs) humming away," McCune says. "We were in this little corner looking like a washing machine."

The rest of teletext's computer operation was no less tentative at that point. There were a handful of VT100s, a standard terminal used to talk to DEC computers, and three or four Norpak IPSs (Information Provider Systems used to create NAPLPS graphics).

"They are based on DEC PDP11/03 computers running the

RT-11 operating system," McCune explained. "The package Norpak provided included two 8-inch floppy disk drives in a freestanding table. The application software which does the graphic creation includes a file transfer program for use between the IPS and RSX or VMS (in capability mode) system."

McCune spent most of his time in the summer of '82 writing macros to add all of the features the editorial staff needed to EDT. That included copy fitting and adding refer (database

Software Tools for RSTS/F.

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, CPUefficient 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



linkage) text. Some two thousand lines of EDT macros were added by McCune which define multiple text buffers and allow for such variations as a small graphic on a screen full of type.

"Most people who use EDT wouldn't recognize it," McCune said. "The question always was, can you make it do this?

"For example, shifting the left margin away from the left hand edge of the screen to accommodate a graphic. I can now sit down and write something to do that in five minutes or less. At the time it took a couple of days.

Essentially, EDT macros are not very powerful. There's no way to branch. There are no conditionals. There's no way to assign values to symbols. Given that very simple operations like counting are difficult, EDT wasn't ever meant to do this. That's not what it's for."

Infomart had written a preprocessor and post processor based on functions specs written by Spohn and McCune. EDT was imbedded between the processors.

So all that remained to be solved was "the main problem of doing the database management and transmission," McCune said.

The software design was done entirely at Informart by a team of five. But it was not your average software project by a longshot.

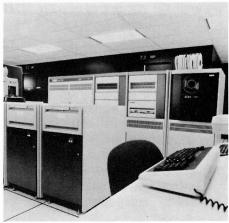
"We said, 'here's a first cut on a functional spec based on what we know now, which is damned near nothing,' " McCune said. " 'As you work on it, we'll keep in touch and we'll be constantly changing what's in the spec.' "

McCune talked to the leader of the software team almost daily for months. Sometimes the calls ran to four-and-a-half hours.

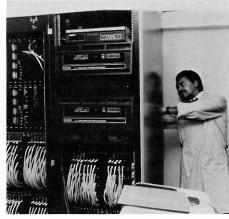
"I'd call Chapman and say, 'I hope you didn't finish that part yet because we want to change it.' "

McCune said the fact that 50,000 lines of FORTRAN were written between March and October of '82 is "astounding," especially given the size of the project and the team.

"Chapman knows the value of having a small development team, even though it was a very big project," McCune says. "One of Doug's lines is



The Codex Room, center of the data communications facilities. All lines between the New York office and editorial room and the computers located in the Meadowlands, New Jersey, pass through here.



Operations engineer Reginald Williams checks the status of the data communications equipment in the teletext offices in the Time & Life Building, New York.

that Jesus had a development team and he had one guy too many, so you certainly want less than twelve."

The working release of ITSS/T came two days before teletext was to go on line in its tent markets in San Diego, California and Orlando, Florida at the end of October.

There have been two modifications since then, primarily bug fixes. Times owns the US rights and retains exclusive rights to some parts of the program. The company has licensed it to its two partners, the San Diego Union and the Orlando Star-Sentinel.

At the end of December, Time teletext added a 780 to its system. The 750 was set aside for development.

"We needed more parts and more memory," McCune said. "We were also getting complaints from the staff that response time was slow. Plus we needed more disk space."

Teletext had been using three RM03s with two megabytes of memory. It now has its 780 hooked up to one RM03, three RM80s and four megabytes of physical memory.

"The RM80s speeded up the disk I/O," McCune says. "That was a problem on the RM03s."

The VAX, then, now drives teletext which often has as many as 30 users simultaneously.

It's just part of a network of computers.

Incoming into the VAX are: a dataline delivering sports and weather realtime information from Bedford, Massachusetts into a DMR using DDCMP protocol. Another line delivers realtime financial information from a New York service called Monchik Weber, also using a DMR with DDCMP. The news wire services Associated Press, United Press International and the French Press go into separate DZ ports. Finally, TV listings on a synchronous link.

There's an asynchronous link using Norpak communications protocol to a Norpak full channel video inserter (FCI).) Teletext bought serial numbers one through four of them. One is used in New York operationally. The locals each use one.

Teletext Incoder System-2, as it's called, encodes the digital data into an analogue video signal. That machine resides in an HBO facility in Manhattan. The signal goes out onto a satellite and is received in cable heads in Orlando and San Diego.

DECNET allows the 750 and 780 to talk to each other for such functions as mail messaging among staff members. Aside from the VTs, the only additional hookup is a line printer.

"Our use of the VAX is unusual," McCune summed up. "It's a good choice for a teletext service. But for the functionality we want, the 780 is overkill. You should be able to provide all of the functionality we have with a 750 or even a 730 if a lot of the editorial processing is done off-line on intelligent terminals like an adapted DEC PC."

McCune says if he had the chance to do it over again, "a distributed process is absolutely the right way to do it."

"So this probably will be the first and last 780 to do a pure teletext service."

With the Nyplan spreadsheet modeling package, the business planner who knows the problem can solve the problem—personally. Nyplan is so easy to use that managers quickly lose any concern over their ability to use it.

Nyplan is designed for both the beginner and the accomplished computer user. You can see what you want taking shape before your eyes—quickly and easily.

Beyond the familiar spreadsheets

and graphics, Nyplan puts at your fingertips a complete financial and statistics capability: Internal Rate of Return, Depreciation, Present Value, Discount; and Trend Analysis, Risk Analysis, Goal Seeking; plus file interfaces.

Nyplan was a financial modeling package for eight years. Now it's a spreadsheet too. The full capability is there. So is the ease of use.

Make us prove it.

ALL RIGHT! PROV information and name of	nearest dealer.
Mail Nyplan, Inc., 12525 V to: Kirkland, Washington TX 705802 NYPLAN L	98033, 206/821-6800
NAME	
TITLE	
COMPANY	
ADDRESS	
CITY/STATE/PROVINCE	
ZIP/MC COUNTR	·



Nyplan gives you more – and makes it easy!

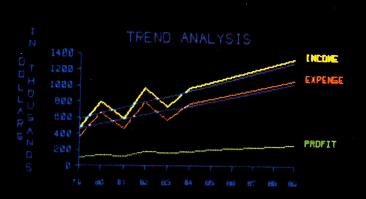
TREND ANALYSIS
WITH LINE GRAPH

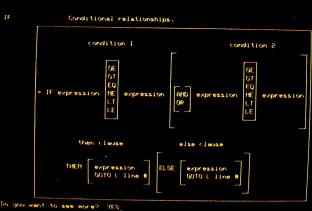
RISK ANALYSIS
WITH PROBABILITY DISTRIBUTION

SPREAD SHEET
WITH SPLIT SCREEN & LOGIC WINDOW

ON-LINE HELP & TEACHING AIDS INTERNAL RATE OF RETURN

WITH BAR GRAPH







David Spohn, now teletext's systems coordinator, agrees.

"The 780 is a powerful machine," he said. "But when you get more than twenty users, it slows down. We have more than that every day."

The VAX, as it turns out, is among the premiere computers in the teletext/videotex industry.

Knight-Ridder's Viewtron is the only US service among the major players that doesn't use the VAX. (A

Viewtron spokesperson said the company uses as Tandem.)

The Times Mirror Co.'s Videotex America uses a VAX, as does NBC's teletext project.

"The 750 is spectacularly reliable," says Richard Reich, NBC teletext's computer system manager. "I've worked in a place that uses 780s and they acted more like computers — they break down occasionally.

"Our 750 was up for eleven weeks

once and the only reason it went down was that I booted it to set a parameter. I had never seen a computer up so long in my life."

Surprisingly, though, reliability was not why NBC bought the VAX to drive its 80-page VBI (Vertical Blanking Interval) service.

It was simply a practical consideration.

"It (the purchase of the VAX) was specified by the technical supplier," Reich, who considers himself an expert on VAX equipment, said Videographic Systems of America, an American corporation held by four French companies and the French government is the supplier. Reich actually is employed by Videographic.

NBC's VAX was installed last February at NBC's computer center in Briarcliff, New York, 40 miles north of Manhattan.

Reich said the software system, written in France according to specs developed by VSA and NBC, is in the final stage of a four-stage installation.

"We have a working system but it's not the final system," he said.

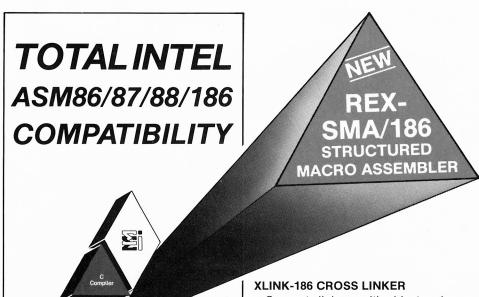
The hardware hooked up to the 750 includes: two RMO5 disk drives with 600 megabytes; a tape drive and some communications gear; some ASCII terminals and three on-line AT&T SCSs (frame creation terminals similar to the NORPAK IPSs mentioned above); the broadcast and insertion equipment.

The NBC system works like this: The VAX transmits pages to two diffusers (VS-D10s). The diffusers hold the pages in their own local memories.

"It's just dedicated special purpose micros," Reich explains. The machines ship out the pages to a device that inserts them into a system in a cyclical fashion (a multiplexer). There are 80 pages an hour in the cycle.

NBC has no decoder boxes in homes so there are no viewers. But technically, anyone who receives NBC would be able to get the service if they had a "black box." Viewers would need a box to unscramble the signal that rides on the "underwaves" or vertical blanking intervals, of the normal broadcast signal.

Editor's Note: Time, Inc. has decided not to roll the service out in 1984 as planned.



REX-SMA/186: A structured macro assembler package for INTEL 8086/8087 and 80186 microprocessors. Includes a macro assembler, an object code linker, an absolute object code locator and an object code librarian. Compatible at command language, source code, relocatable and object code levels.

MULTI-USER, MULTI-TASKING CAPABILITY OF DEC MINICOMPUTERS

Takes advantage of the versatility and power of the DEC PDP-11 and VAX series mini-computers. Frees up your INTEL MDS for software/hardware integration testing.

XASM-186 CROSS ASSEMBLER

- Supports 80186 extended instruction set.
- Supports 8087 co-processor code development.
- Supports INTEL compatible Macro Processing Language facilities.
- Produces INTEL compatible relocatable object module format.

- Supports linkage with object code produced by any INTEL language processor.
- Supports incremental and overlayed program linkage
- Improved link map output with sorted symbols.
- Supports type checking of global variables.

XLOC-186 CROSS LOCATOR

- Produces INTEL compatible MCS-86 absolute object code format.
- Useable with SoftProbe language oriented software simulator or INTEL's ICF
- Improved memory map output with sorted symbols.

XLIB-186 CROSS LIBRARIAN

- Supports library management of INTEL compatible object code modules.
- Uniquely enhanced by library module replacement and individual module extraction from the library.



(714) 241-8650 TWX: 910-695-0125 ©1983 Systems & Software, Inc. SSI 83-6

RESOURCE ACCOUNTING UNDER VAX/VMS

PART 1

By Howard Reisman

Editor's Note: For an explanation of what resource accounting is, and specifically, how it is performed under VAX/VMS, we asked Howard Reisman, president of Computer Information Systems, Braintree, Massachusetts, to write this two part series. Here in Part One, Mr. Reisman explains what resources VMS measures and how, as well as the format of the accounting record. Part Two will conclude with controlling the accounting file, user defined accounting records, the ACCOUNTING utility, and thoughts on image level accounting.

WHAT IS RESOURCE ACCOUNTING?

Resource accounting is the capture and recording of data that indicates the utilization of a computer system by its users. This article will explain how it is performed under VMS.

WHAT RESOURCES DOES VMS MEASURE AND WHAT DO THEY MEAN?

The most commonly measured resources on computer systems are:

- 1. Connect time which refers to the amount of time your process was logged into the system. As we will see later, it is actually a derived value under VMS, computed by subtracting the start time of your process from its finish time, both of which are stored by VMS. The units are measured in hundredths of a second.
- 2. **CPU** time which refers to the amount of time the VAX processor is actually executing on your behalf while you are logged in. CPU time is also measured in hundredths of a second.
- 3. Disk storage which refers to the number of 512 byte blocks maintained on line on your behalf. As we will see later, this information is maintained separately from the other resources I have mentioned. Disk storage is also maintained by UIC, whereas the other resources are maintained by username. If there is not a unique username to UIC relationship on the system, the disk storage by UIC will reflect the sum of disk usage for each username with that UIC.

The operating system is the software component that is given the task of accounting for and recording resource usage. Each operating system performs resource accounting differently. Those of you who are familiar with both RSTS and VMS are aware of the completely different approach to resource accounting taken by each of the two operating systems, both in the resources that are accounted for and in the method of storing the resource data.

Now let's consider the resources that VMS accounts for. In addition to the three basic resources, there are several others that are measured under VMS:

4. Page faults refers to the number of times your process referenced an address that belonged to a page that was

not in your working set. The page fault generates an exception that causes the pager to run in your process context. The pager locates and adds the faulted page to your working set, so your process can continue execution.

- 5. Page fault I/O count refers to the number of times the pager actually had to go to disk to retrieve the faulted page. These faults are considered expensive because the page fault results in a disk I/O request, which slows down execution of your process. Most page faults turn out to be "cheap" faults, since they are satisfied by pages already in memory. Pages found on the free and modified list, and global pages that are not in your working set, but are in someone else's working set all fall into this category. Demand zero page faults also do not result in disk I/O. These factors should result in page fault disk I/O being substantially less than the number of page faults actually generated.
- 6. Buffered I/O refers to the number of buffered I/Os that were performed by the system due to requests from your process. Buffered I/O is an I/O request where data is transferred between a peripheral device and intermediate buffers in the nonpaged system pool. The data is then transferred from the system pool to your process buffer. Buffered I/O is more expensive than direct I/O since it uses an intermediate system buffer and actually results in an extra memory to memory transfer. Usually buffered I/O consists primarily of terminal I/O. However unspooled line printer I/O, mailbox I/O and DECNET I/O also result in buffered I/O. Buffered I/O is generally used for "slower" peripherals.
- 7. **Direct I/O** refers to the number of direct I/Os that were performed by the system due to requests from your process. Direct I/O is an I/O request where data is transferred directly between a device and your process buffer. Direct I/O does not require the nonpaged system pool as an intermediate buffer. Direct I/O is primarily thought of as disk I/O, however magtape I/O also qualifies. Direct I/O is used for "fast" peripherals that do large DMA transfers. By the way, you do not get charged for any ACP I/O that occurs "behind your back" as a result of your I/O request. The ACP is charged for all "behind the back" overhead, such as file opens and window turns since it is a separate process.

- 8. The number of volumes mounted refers to the number of times a volume on a device was successfully mounted by your process. Volume mounts can occur for disks, floppy disks and tapes.
- 9. The number of pages printed refers to the use of the symbiont print queues by your process. The number of pages printed is a computed figure. It is based on the number of lines printed and the page size setting used for the printer, defined by the DCL command SET PRINTER 'printer'/PAGE=. If you allocate a line printer and then output directly to it (unspooled output), the number of pages printed is not maintained. Pages printed is only maintained for spooled printing via a symbiont. There is currently a problem with VMS in computing pages printed when the page size is set to zero. You get some awfully strange looking symbiont page counts.
- 10. The number of images executed is simply a count of the number of image activations performed by your process. Image activations are relatively expensive on VMS, so a high number of these can be a source of a great deal of system overhead. Remember, most DCL commands result in an image activation. Command files often do a great number of image activations behind your back in order to achieve their sometimes magical results.
- 11. The peak working set size refers to the maximum size that your process or image grew to in pages while executing. This refers to real memory consumed by the process or image. Note that the use of global pages counts as part of your working set and consequently your real memory usage, even if the global page is also in use in a number of other process working sets at the same time.
- 12. The peak page file usage refers to the maximum number of pages your process or image maintained in the system page file while executing. The pages that are maintained in the system page file are the demand zero and copy on reference pages that were modified during process execution and do not fit into your working set. Since there is also no room for these pages on the modified page list, they are forced to secondary storage.

HOW ARE THESE RESOURCES MEASURED?

To simplify this discussion a bit, let's exclude disk storage for the time being, since the accounting and storage mechanisms used for disk storage are completely different from the other resources.

The VMS operating system maintains a system accounting file called SYS\$MANAGER:ACCOUNTNG.DAT. This file stores all the VMS resource information mentioned earlier (excluding disk storage) for all users. While a process is executing, VMS maintains process resource accounting information in the fixed portion of the process header. This information is updated as it happens. When a process terminates, VMS writes a process termination record to the end of the accounting file. Resource accounting records can also be written when other events occur. Records are written chronologically to the end of the file as events occur. The following is a complete list of the events that can cause a resource accounting record to be written to the system accounting file:

1. A process termination

- 2. An image termination
- 3. A system initialization
- 4. A login failure
- 5. A print job termination
- 6. A user program sending a record to the system accounting file
 - 7. Opening a new system accounting file.

Process terminations can be divided into the following categories:

- 1. Interactive processes are what we typically think of in using the computer. They refer to when a user logs into the system, does some work and then logs off.
- 2. Subprocess terminations occur when a process initiates a subprocess and that subprocess later completes execution and is deleted. A subprocess shares certain resource quotas between it and its creator process, known as its owner. For example, the open file limit quota is shared. A subprocess must terminate before its owner does. Subprocesses can be started in a number of ways. From DCL level, the SPAWN command does the trick, as well as the RUN command with any qualifier (except /UIC which creates a detached process). Within a program the \$CREPRC system service can be used to create a subprocess. A call to the run time library module LIB\$SPAWN will also create a subprocess.
- 3. **Detached processes** are independent processes that another process can start if the other process has the DETACH privilege. Detached processes can be created either from DCL level with a RUN/UIC = command or within a program with the \$CREPRC system service. Detached processes have no process relation to the creating process, unlike subprocesses which are owned by the creating process.
- 4. **Batch jobs** refer to command files that are submitted to the batch facility for either background or deferred execution, freeing the terminal for other tasks.
- 5. **Network processes** refer to processes that were created due to a connect initiate request from another node (or the same node) in a DECNET network. Network file transfer and process task to task communication both create network processes on the destination node.

Disk usage is maintained differently. As mentioned earlier, it is maintained by UIC, not username as other resource information is. Also there is no single central location where disk usage information is maintained. Each disk maintains its own information. To get a complete picture of system disk usage, each disk of the system has to be examined and data must be accumulated and summed. In the case where multiple physical disks are bound into one logical volume, only the root volume (volume 1) of the volume set need be examined.

Disk usage information is available from two separate sources on the disk. The first is the system index file [000000]INDEXF.SYS. This is a file which maintains information on all files on the disk (among other things). One of the items of information stored for each file is the owner UIC of the file. The number of blocks allocated for the file is also stored. The INDEXF.SYS file is updated as events affecting disk space allocation occur. The file can be read and storage information can be extracted via either a user program that reads INDEXF.SYS or the DCL command

We Have the Prescription for Your Ailing Computer System



Tymshare's computer maintenance organization is just what the doctor ordered.

We're an independent service group offering you fast response time and rapid, reliable repair—24 hours a day, seven days a week.

Maximum Uptime for the Lowest Possible Cost

Our highly skilled, experienced service representatives quickly and accurately diagnose your

system's problems—to maximize your computer uptime.

And our custom preventive maintenance program keeps your installation healthy—to ensure continuing efficient operation at the lowest possible cost.

For more information on how Tymshare's maintenance programs can best fit your needs, call us today at the regional office nearest you.

COMPUTER SYSTEMS and SUPPORT

EAST: 10111 George Palmer Hwy.; Bowie, MD 20716; (301) 459-8363
CENTRAL: 4513 Lincoln Ave., Suite 106; Lisle, IL 60532; (312) 964-0248 or 964-0362
SOUTHERN: 6333 Harry Hines Blvd; Dallas, TX 75235; (214) 638-2130
SOUTHWEST: 11050 E. Artesia Blvd., Suite D; Cerritos, CA 90701; (213) 402-1114
WEST: 39100 Liberty Street; Fremont, CA 94538; (415) 794-2490 or (415) 794-2719



ANALYZE/DISK/USAGE "diskname." This disk usage information has to be further summarized by UIC, since the data is returned by file, not by UIC.

Disk storage information is available in a more convenient form via the utility DISKQUOTA. The system uses the file [000000]QUOTA.SYS to maintain disk quota and disk usage information for that disk. This information is summarized by UIC. DISKQUOTA can read and display the current state of the QUOTA.SYS file. To use DISKQUOTA, disk quota checking must be enabled for the disk. This adds some additional system overhead since every time a file is created or extended, the disk quota file must be consulted and then updated.

THE FORMAT OF THE ACCOUNTING RECORD

Let's now consider the structure of the accounting file in depth. The accounting file is an RMS sequential file with variable length records. The owner UIC of the file is [1,4] and the system default protection code is used, which is determined by the SYSGEN parameter RMS__FILEPROT. The file is stored on the directory SYS\$MANAGER:, which on our system translates to DRAO:[SYSO.SYSMGR]. The name of the file is ACCOUNTNG.DAT. The system keeps the highest version of this file opened so that events that generate accounting records can be written to the file as they occur. The events that can generate accounting records are:

- 1) Process terminations
- 2) Image terminations
- 3) System initializations (i.e. booting the system)
- 4) Login failures
- 5) Printer job completions
- 6) Sending user defined messages via \$SNDACC
- 7) Creating a new accounting file

All accounting records are immediately written to disk when they occur, except for image terminations records, which are buffered and written in a clustered fashion so as to minimize excessive disk I/O to the accounting file. Image records are flushed when another record type must be written to the accounting file.

The accounting file currently stores eight different record types, corresponding to the termination events previously listed. They are as follows:

- 1. A process termination record is written when an interactive process, subprocesss, detached process, batch process or network process terminates. The information contained in the record indicates virtually everything there is to know about the process. We will consider the exact format of the stored resource data shortly. The stored data could be classified as belonging to one of two groups, process state information or resource usage data. Process state information includes username, UIC, account, terminal, process ID, priority and privileges. Resource usage data includes start time, finish time, CPU time, page faults, page fault I/O reads, direct I/O, buffered I/O and number of images executed.
- 2. An image termination record is created when an image exits. The actual writing of the image record to disk is generally deferred somewhat due to buffering. Image records are identical to process records concerning the process state and resource usage information maintained. Addition-

ally the name of the executed image is stored. Image level accounting is the only type of accounting that is not enabled when the system is brought up, due to the relatively large CPU and disk storage requirements needed to support image level accounting. Rather than accumulating and writing out process state and resource usage statistics at each process termination, a relatively infrequent event, information is accumulated and written out at the end of each image termination. The number of images executed in a terminal session or batch job can be considerable, especially on machines used for program development. Complex command procedures can also perform a huge number of image activations in a short period of time. Also almost all DCL commands result in the execution of an image.

- 3. A system initialization record is written for the process that runs the system startup command file when bringing up VMS. This record is identical in structure to a process termination record. The username for the system initialization process is SYSTEM and the UIC is [1,4]. These records indicate when the system was brought up, how long it took, and the resources required by the system initialization process to bring the system up. A login failure record is written when any unsuccessful attempt to access the system occurs. The information provided in the login failure record indicates the username and terminal from which the unsuccessful login attempt occurred as well as the usual process state and resource usage information. No information is given regarding the password used for the attempted login. Also if the username is either syntactically invalid or is not set up in SYSUAF.DAT, it is not stored in the accounting record. The string < login > is stored instead. Obviously the system manager should be concerned if a number of login failures have occurred in quick succession, especially if they occurred off hours, on dial in lines or for either nonexistent users or known user accounts (e.g., SYSTEM). If security is an issue at an installation the login failures should be examined periodically, perhaps even daily, for occurrences that may exceed the expectations of users mistyping their usernames or passwords. It is also important to look at any successful logins after a series of suspicious attempts. These may indicate actual system penetration.
- 4. A print job termination record is written when a print job sent to a symbiont print queue completes. The information contained in the print job record differs somewhat from the process or image termination record. A print job termination record maintains the physical queue name the job was printed on, which is not necessarily the queue the job was sent to. For example, jobs queued to SYS\$PRINT get changed to the physical queue name (e.g., LPAO). The job name and the queue entry number are also maintained, as well as most of the process state information mentioned in the process record. The resource usage information maintained is less than what is maintained in the process termination record. Print job records only maintain the queue name and finish time for the job, the number of QIOs issued, the number of RMS gets performed and the number of pages printed. As you will see later, there is storage reserved in the accounting record for the start time of the print job and the amount of CPU time the symbiont used to print the job, but currently these fields are not used.

5. A user defined accounting record is a record that a user program sends to the accounting file via the \$SNDACC system service. No privilege is required to do this. The user may send from 0 to 255 bytes of data per accounting message. The form of these bytes is completely user defined. It is also up to the user to process, decode and ultimately use the data written in a user defined accounting record. VMS will preface the user defined data with process context information. I will give an in depth example of writing user defined accounting records later in this article.

6. Forward and backward file pointer records are used to indicate logical links between accounting files. File pointer records are created when the DCL command SET ACCOUNTING/NEW_FILE is issued. When the old accounting file is closed, a forward file pointer record is written to the old accounting file which points to the new accounting file. The forward file pointer is the last record written to the old accounting file. Correspondingly, the first record of the the newly created accounting file is a backward file link record, pointing to the old accounting file. The forward and backward file pointer records indicate the date and time the new accounting file was initialized and opened.

We will now consider the structure of accounting file records in depth. The accounting file symbolic constants that are about to be mentioned represent either bit or byte offsets to fields in records or actual values. The constants themselves are integer values. They are defined in the MACRO \$ACRDEF which lives in the MACRO library

SYS\$LIBRARY:STARLET.MLB. Constants can change in future major VMS releases. Consequently when programming, it is always a wise idea to use and link with the symbolic constants rather than the integer values they represent. The values given for the symbolic constants are for VMS version 3 only. A little further on, a sample MACRO program will demonstrate how to get at and link with symbolic constants defined in the macros that are stored in STARLET.MLB.

All accounting file constants begin with the prefix ACR\$ followed by a letter that indicates the field type the constant represents. The constants themselves are always integers. The data type indicator is followed by an underscore which is then followed by the unique name for the constant (e.g., ACR\$W_TYPE). The following chart may prove helpful;

Data type	Meaning	Field type	Example
V	Offset	Bit group	ACR\$V_PACKET
В	Offset	Byte integer	ACR\$BPRI
W	Offset	Word integer	ACR\$WLENGTH
L	Offset	Long integer	ACR\$LPID
Q	Offset	Quadword	ACR\$Q_SYSTIME
T	Offset	Counted string	ACR\$TIMAGENAME
K	Value	Integer	ACR\$K_LOGFAIL
			continued on page 60

BACmac can do it all!

BAC into RTS / BAC into MAC / BAC into BAS



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

CIRCLE 138 ON READER CARD

FOR RSTS/E VT.1, 7.2 8, 8.0

RPM CAN DRAMATICALLY INCREASE THE PERFORMANCE OF YOUR SYSTEM

You can double or triple your system's performance without adding costly hardware. RPM provides everything you need to optimize your system and keep it running at peak performance.

RPM analyzes your system performance automatically, identifying problem areas. But, RPM doesn't stop there. It identifies the cause of the problems and makes suggestions for correction in plain English.

AUTOMATIC PERFORMANCE ANALYSIS

Instead of dumping columns of cryptic numbers, RPM gives you a plain English report that describes how your system is performing. It tells you where you have problems, what caused the problems, and how to fix them.

This report analyzes each of the resources that are common problem areas. Any resources that are not being used optimally are identified. The program or files that caused the problem are then identified and suggestions are made for correcting the problem.

DETAILED PROGRAM ANALYSIS

In addition to identifying problem programs, RPM can analyze the operation of individual programs, identifying problem areas.

RPM's detailed program analysis breaks the operation of the program down into CPU usage, input/output and system calls. Usage and directory overhead counts are displayed by channel and by system call.

---- File Processor (FIP) Usage ----

** The file processor (FIP) is being excessively used. It is in use by at least one job 67% of the time. In addition, an average of 2.7 other users are waiting to use FIP. Although the file processor is in use 67% of the time, it is waiting for information from the disk 72% of the time it is in use. The disk information that FIP is waiting for breaks down as follows:

se. The disk information that 1.
91% Directory.
Reading or writing to the directory structure.
7% Disk Allocation Table (SAT)
Reading or writing information about free blocks on the disk.
2% Monitor Overlays.
Loading monitor overlays into memory.
0% Miscellaneous.

U% Miscellaneous.

Loading disk cache information and other miscellaneous data.

The five programs that use the most FIP resources are:

0E047A 275.2

PAYREC 213.1

...TKB 158.0

BP2COM 110.3

LOGIN 0.7

LOGIN

FIP usage can be decreased by optimizing the clustersize of frequently used files (see section 4.4.1 of the RPM User's Guide), using contiguous files where possible, reordering the directories often and minimizing the opening

RPM> EXAMINE JOB 5 EVERY 5 MINUTES

Job:	5 P	rogram:	*ALL	* CI	PU: 54	% Sam	nple T	ime:	297 Sec	onds		
Chan		File	•	C	ount 0	vrhd Ch	nan		File		Count	Ovrh
0 K	B2:				3		1 * 0	losed	*		4	
3 DI	M1: [1,	3]CSPCC	M. OLB		835	2	4 DM1	: [5, 1]	TEMP05.	TMP	39	4
6 D	M1: [5,	1 EXAME	L. STB		9	91	7 DM1	: [5, 1]	EXAMPL.	TSK	733	
15 D	M1: [1,	3] TKB	. TSK		278							
EMT	Count	Ovrhd	EMT	Count	Ovrhd	EMT	Count	Ovrhd	EMT	Count	Ovrhd	
CALFII	149		. READ	1441	2	. WRITE	460	133	. CORE	7		
. TTECI	1 1		. TTRST	Γ 1		. DATE	1		. NAME	2		
. RTS	1		. LOGS	12		. CLEAF	2		. CCL	1	5	
. FSS	42		. UUO	71		. RSX	1		CLSFQ	62	28	
OPNFQ	58	535	CREFQ	2	226	DLNFQ	1	23	RSTFQ	2		
LOKFQ	22	259	CRTFQ	1	64	CRBFQ	1	175	UU. ATF	70	250	
UU. NA	4 1	3										

EXTENDED PERFORMANCE STATISTICS

RPM adds extended performance statistics to your RSTS/E monitor. This monitor extension captures information about overall system performance plus information on individual programs and files. Using this information, you can improve system performance by minimizing disk head movement, reducing file processor (FIP) waits, reducing swapping, and optimizing disk cache operation.

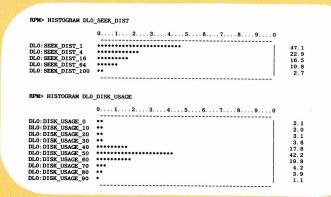
DYNAMIC PLOTTING

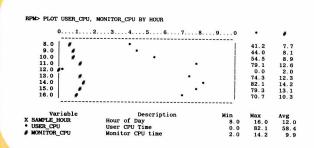
In addition to comprehensive reports, RPM can generate a wide variety of graphs. It can plot curves, draw bar charts, and plot histograms using any combination of system information. By plotting one variable against another, you can immediately see correlations between variables. This is espescially useful for determining critical resource usage points.

IDENTIFY PROGRAMS BY RESOURCE USAGE

With RPM, you can identify the programs that will provide the most benefit from optimization. You can determine which programs are used most often and which programs use the most of critical resources.

Once identified, these programs can be optimized. Overall system performance can be increased by making changes only where they will do the most good.





```
IST TOP 5 PROGRAMS BY PRG_CPU_TICKS
PARREC
DAYEND
PAYROL
FAIAND 103
CLEENUP 73
1ST TOP 5 FILES BY FILE_DIR_OVRHD
13, 10] PROFIL JOU 33.1
13, 10] PROFIL BZS 27.6
[2,0] MTHEND. TSK 19.6
[10,1] EOWREC. JOU 11.3
[2,0] PAYAUD. DAT 7.8
```

NO-RISK GUARANTEE

If all this sounds too good to be true, here's your chance to find out for yourself. Try RPM for 30 days. Use it to tune your system. If RPM isn't everything we say it is and more, return it for a full refund. We're not worried. We know that once you see the difference RPM can make, you won't want to be without it.



RPM -- The Performance Revolution

NORTHWEST DIGITAL SOFTWARE

AN ALTERNATIVE TO DATATRIEVE

By Ann Sangpan, North Shore Sanitary District, Gurnee, Ill. Editor's Note: "An Alternative to Datatrieve" was presented at the Spring DECUS symposium. Ms. Sangpan has updated her report for our readers.

From time to time, users have come to our department seeking help in creating files for storing data. They wanted to be able to sort and conditionally select the data, and print their selections in a report. For instance, people in the Lab Department needed files to store their test results or the Personnel Department needed files to store temporary employee records. As a result of their expressed need for a free hand at developing their own applications, we designed a 10 hour Datatrieve course for the users. To some extent, Datatrieve was very effective in meeting the needs of some of our users. Those who frequently use Datatrieve became accustomed to the quirks of the system, memorizing the syntax and commands. However, those who only occasionally used Datatrieve found it difficult to work with. They either had to look commands up in the manual or ask us for help (they usually did the latter). Then we purchased Digital's Application Development Environment (ADE) as an alternative to Datatrieve. ADE is a package that can be used to enter and store information. By using ADE commands, changes can be made to the data, calculations can be computed, and reports can be generated.

The system is designed to help the user, who has little or no computer experience, with record keeping or bookkeeping tasks. ADE is the perfect package for upper management who does not have time to learn all the intricacies of a reporting system. Users who infrequently use the computer will also find ADE's step-by-step instructions easy to follow and understand rather than Datatrieve's prompts. Prompts in Datatrieve such as "Looking for a Boolean expression" are rather confusing to an inexperienced user. Before I continue further, however, I would like to assure wombat-lovers that I am not suggesting that ADE can replace the use of Datatrieve. What I am suggesting is that certain users may find ADE easier

to use than Datatrieve.

CONCEPT

When a non-data processing user needs a quick report, he or she needs an understandable package to assist in completing the desired task. Although both ADE and Datatrieve can provide that report, the concepts in Datatrieve are more on a data processing level than ADE.

The main concepts used in either system involve how information is stored in files.

DATATRIEVE:

DOMAINS

RECORDS

FIELDS

ADE:

TABLES

LINES

COLUMNS

In Datatrieve, the collection of records is called a DOMAIN. ADE uses a more familiar concept when dealing with data files. In ADE, TABLES are used as data files. Most users have, at one time or another, worked with tables of some kind. Thus, the idea of storing information in tables is easier to understand than storing information in domains.

Each grouping of related data is called a record in Datatrieve. The record is made up of fields. To the inexperienced user, the relationship between the record and its fields may not be clear. Because a user is more familiar with tables, it is easier to understand that a line in a table is a record. The fields in ADE are simply the columns in the table.

Command files are available through ADE and Datatrieve. However, ADE provides the user with two kinds of command files.

DATATRIEVE:

PROCEDURES

ADE:

VERBS

REPORTS

Procedures which produce a report, are called REPORTS, while those

which are simply commands, are VERBS. Since many users find command files difficult to understand in the first place, the two kinds are distinguished. The idea that REPORTS generate reports and that VERBS contain commands is clear to the user. Datatrieve has PROCEDURES which can contain commands and/or reports. A user may be uncertain about when to use a procedure.

CAPABILITIES

ADE and Datatrieve have many of the same capabilities. Both allow the sorting, adding, changing, and deleting of records in files defined in ADE or Datatrieve. The two systems have the main capability to select certain records and generate fast reports with totals and calculations. Small programs are written in ADE (verbs) or in Datatrieve (procedures) which processes a group of commands when invoked.

Aside from the obvious differences in commands, differences in capability are also found between ADE and Datatrieve. One major difference between the two is how files are created. To create a file in Datatrieve, one must have a certain knowledge of data files, its organization, and the data which is to be stored in the files. The size of each field must be decided before all the information is stored. In ADE, the user does not have to worry about the size or organization of a file. ADE automatically expands the columns as needed when information is entered. When defining a table in ADE, the user is prompted for the key by which he would like the file to be sorted.

Another significant difference is the flexibility of the two systems. Because a user often does not know what information he needs until he sees it is not available, ADE provides a flexible system to store data files. Whole files can be copied without exiting ADE or specific columns can be copied to another table. Using ADE defined commands, tables can be merged, appended, or combined. A backup command is available to restore a worksheet to the original condition at the beginning of the session. This is very helpful in case a user changes his

worksheet accidentally, or if he changes his mind (and we know this happens often) as to what kinds of modifications he wants to make.

When a user needs another piece of information in a certain report, ADE allows the user to add a new column to the table. It will print the needed report and have the old table stored without the new column added. Of course the new column can be saved if the user decides that the field may be needed in the future.

Both ADE and Datatrieve are capable of generating reports. Reports can be created from a selected group of records or print selected fields. ADE, however, provides a capability to convert a group of data into a graph form. The user does not have to manually convert data into a graph. Using a command in ADE, two column names are prompted for, one of which must be numeric. A report in the form of a graph is then created, either to the user's terminal or to a specified report name.

	Bar	graph	of QTY	versus	PART	rs
250	!					
	!				X	
200	!	Х			X	
	!	Х			X	
150	!	X			X	Х
	!	X			X	X
100	!	X			X	X
	!	Х	Х		Х	X
50	!	X	Х		X	X
	!	X	Х		X	Х
	WID	GETS	GADGETS	FOBL.	FS (TZMOS

There are many capabilities which Datatrieve has which ADE does not. Where ADE is restricted to only 1000 records in a table, Datatrieve files can store a larger number of records in its files. Datatrieve can store data with multiple occurrences in lists. Datatrieve files can interact with system files while ADE files cannot. ADE also executes slower than Datatrieve. It must be kept in mind, however, that those who use ADE are not computeroriented. Most likely. ADE users do not need to use large complex data files or to access system data files. Although the speed of producing a report may be important, a user will find that using ADE is quicker than manually compiling a report. The extra few seconds in ADE are not as noticeable to a noncomputer user.

COMPLEXITY

ADE provides step-by-step

prompts that tell the user what types of information to enter. If the user is familiar with ADE, he can type all the necessary commands needed to complete the taks. Thus, he is not prompted for them. At any step, the user can call the help mode to find more information on the commands for which he is prompted. ADE will specify if a command (GET, MAKE, CHANGE) is needed or if it needs a subject (TABLE, RE-PORT, VERB). Below are the first prompts a user sees when he accesses ADE and Datatrieve.

Type HELP for help
DTR>

Ade:

Enter a Command, a Verb,
Report name, or type HELP

It is clear how the prompts in ADE may help the inexperienced user complete a task. Another example of ADE's step-by-step instruction is seen when

"Did I really have to print all these files to get the few pages I needed?"



Before

- I used to pour over printouts for hours to extract data.
- I reprinted entire files to reproduce lost invoices.
- I wasted money on tons of paper that no one read.

Then I Found READER

A software package designed expressly to

 R_{ead} , E_{xtract} A_{nd} D_{isplay} R_{eports}

Now

- I can interactively display and extract the data I require in seconds.
- I can reproduce individual invoices, checks or statements in minutes.
- I significantly reduced printing and paper costs.



You Too Can Benefit from READER™

- Runs on VAX, DEC-10, DEC-20
- Terminal independent
- 80–132 character support

VAX, DEC-10, and DEC-20 are registered trademarks of Digital Equipment Corporation.



CALL FOR FREE EVALUATION

INTERACTIVE SYSTEMS, INC. 131 Middlesex Tpk. Burlington, MA 01803 617-273-4420 Telex 951333 the following commands are entered:

```
|-----|
(prompt)
  DTR>
(command entered)
   "READY"
(response)
  DTR>
!----- ADE ------
(prompt)
   * Enter a Command, a Verb,
    Report name, or type HELP
(command entered)
   "GET"
(response)
   Ade: GET table
   * Enter an existing Table name
```

In ADE, a table name, which is the next information needed, is prompted for, while in Datatrieve, the user must know that a domain name should be entered next in order to complete the READY command. If the user is familiar with ADE, he can type the command to get a table and the name of the table without waiting for the prompts. Datatrieve does have a GUIDE mode which a new user can use to help execute commands. Because it is very restrictive about what commands can be used while in the mode, a new user may be confused when a perfectly valid command is entered and Datatrieve responds with an error message.

The step-by-step prompts in ADE are also present when making a report procedure. Everything required, from the title of the report to the layout of the report is prompted for. Instead of the user having to memorize or look up the components of a report procedure, ADE prompts for each component so the user can simply make a report without being familiar with the package.

ADE's flexibility further simplifies its use. In ADE, existing tables are easily changed in structure. Columns are added or deleted, or the order of columns is changed using ADE commands. It is a more tedious task to add columns, however, in Datatrieve. The record definitions are redefined. The files are copied to a new file with a larger or smaller record size (this can

only be done through the use of several complex Datatrieve commands). Changes to records are also easier to make in ADE. Many times, a user may not know if a record needs changing until he sees it. When an ADE user is changing a record, the existing record is automatically printed above the prompt for new values. Datatrieve, however, must be instructed with a command to print the old record before being instructed with the command to change the values. When changes are made to a procedure in Datatrieve, the procedure must either be redefined, or the user can change it using an internal Datatrieve editor. However, this editor has its own different syntax rules, therefore, an inexperience user will have great difficulty. Changes in ADE can be made to reports or verbs using a command, without redefining the procedure or using unfamiliar syntax.

Aside from the descriptive prompts and ADE's flexibility, it is simply easier to do certain tasks in ADE than in Datatrieve. To create a file, for example, requires numerous steps in Datatrieve. To create a file which stores sales results, the following commands must be entered in Datatrieve:

The user needs to understand picture clauses and computational usage clauses in order to create the record definition. Alternately, the Datatrieve user could get assistance from the Application Design Tool (ADT) in making file domains and record definitions, but once again, not all valid syntax is supported. To define this same file in ADE the following steps need to be entered:

ADE prompts for each column name and asks which name is to be the key field. As data is entered, the size of each column will expand to store the information. After the key selection has been entered, the table SALES is

created. ADE then prompts for input of the data which is stored in the table. If the user does not store the data in the table at this time, the table will still be available to store data whenever the user wishes.

COMMANDS

ADE's commands are more like every-day language than Datatrieve's commands. A command such as GET in ADE versus the READY command in Datatrieve is an example of how the commands can affect the understanding of the system. It is easier to understand that one must GET a data file in order to use it rather than one must READY a data file before it is accessable.

Other commands in ADE such as ENTER, CHANGE, and MAKE, are understandably simpler than STORE, MODIFY, and DEFINE. Although these commands perform the same task, the user who is not familiar with programming terms may be more comfortable using ADE. Commands in ADE which are not in Datatrieve are also very simple to understand. For instance, the MOVE command is used to move columns from one place to another. The PICK command allows the user to pick which columns in the worksheet he wishes to keep as his permanent table.

CONCLUSION

We have found that there are a few advantages in using ADE in particular instances. For one, the 1000 record limitation assures us that our users are not creating huge unmanageable files. Secondly, ADE's simplicity frees us, the data processing department, of the interruptions and questions regarding the creation and maintenance of their data files.

Of course, on many occasions, it is more efficient for a user to use Datatrieve. Indeed, ADE cannot replace every aspect of Datatrieve. However, the different uses and syntax of Datatrieve may overwhelm a user who is not program oriented. Although Datatrieve allows the users to work with larger and more complex files, many users do not need a system with such capabilities. For the user who needs an occasional report or place of storing minimal data, we have found an alternative: ADE.



LOGICRAFT Inc.

3 Graham Drive, Nashua, N.H. 03060, 603-888-4448

INTRODUCTION TO THE VAX/VMS A FIRST TIME USERS GUIDE

By Terry C. Shannon, The Poise Company, Roswell, NM

Editor's Note: With more than a dozen volumes of manuals in the VAX/VMS documentation set to sift through, a new VAX user can totally dry out three pairs of contact lenses just learning to log in! "Front-Ending" the VAX DOCSET at your site with your own, tailored VAX manual can get your users on-line faster, with a lot less help from you. (You can use this extra time to figure out how to run BACKUP or stop a runaway batch job.)

"Introduction to the VAX/VMS — A First Time Users Guide," By Terry Shannon is excerpted below and can serve as a "model" for you.

VAX/11

Your computer system is a Digital Equipment Corporation VAX/11 (Virtual Address eXtended) running under the VMS (Virtual Memory System) operating system. In computer jargon, the VAX is referred to as a virtual memory, multiprocessing superminicomputer. For our purposes, it is sufficient to note that the VAX is an extremely powerful machine: little more than a decade ago, a computer with the capabilities of even a small VAX system filled up a large room and carried a multimillion dollar price tag.

Three features of the VAX permit it to function as it does: virtual memory, multiprocessing, and the VMS operating system. Here's a brief description of how each feature works:

VIRTUAL MEMORY is the ability of a computer to address and use more memory than its central processing unit physically contains. A one megabyte VAX processor can address not just the one million locations contained in its physical memory, but, due to its 32-bit address length, a total of over four billion storage locations. Now, how does a computer make use of four billion addresses when it can store only one million of them at a time? The answer is simple — the VAX does essentially the same thing that you do every time you read a book. Regardless of the overall length of the book, you read it one page at a time. The computer does the same thing — it processes a program page by page. To the VAX, a program is similar to a book composed of many pages which is stored on its disk drive. When you issue the command to run a program, the computer first scans the overall program, breaks it up into pages, and creates an index to reference the location of each page. Each page is a 512 byte, or word, segment of a program which is read into the CPU from the disk and is then processed or acted upon. When the CPU is finished with the segment, it scans its index to locate the next page and then generates a "page fault" by returning the completed page to the disk and then reading in the next segment needed to continue processing the program. In essence, the page fault executed by the VAX is equivalent to the human action of turning the page of a book. Because of its ability to process a program one page at a time, the VAX can process a program of almost infinite length within the confines of its physical memory.

The VAX also has MULTIPROCESSING capabilities,

meaning that it can run programs for and interact with a number of people at the same time. This usage is almost totally independent: each user who is logged into the system can utilize all of the available system resources regardless of what the other users are doing. For all practical purposes, each user has his or her own independent computer system.

What ties virtual memory, multiprocessing and users together is the computer's OPERATING SYSTEM — a set of programs, utilities, system services and device drivers within the VAX that acts as an interface between the computer and its peripheral devices and other resources, and its users. The operating system, called VMS, schedules and controls computer resources, peripheral equipment, and program execution. This takes a large burden off the user who needs only to type in RUN and a program name to execute a program. VMS and the System Manager handle everything else.

1.0 SOFTWARE

Software is best defined as the sum total of programs, programming language compilers, utilities and commands available within a computer system. VAX processors support a wide variety of software. Some of the most commonly used software is listed below:

VAX/VMS — The Virtual Memory Operating System (or system software) which supervises the internal functioning of the computer and its peripheral equipment. You communicate with the VAX by entering commands in Digital Command Language (DCL), and VMS acts as an interface between the computer system and you.

EDITORS — These are utilities which users invoke to create or change programs, text files and other data in the computer. Several different editors are available on the VAX, including EDT, SOS and TECO. The most powerful and flexible editor that you can use with a video terminal is EDT. You can learn how to use the EDT editor through the EDT-CAI Computer Assisted Instruction Program on the VAX.

PROGRAMMING LANGUAGES — A large number of programming languages may be used on VAX systems. A list of some of the more popular languages follows:

VAX BASIC — Similar to, but far faster and more powerful than the BASIC programming language used in home computers. BASIC is probably the easiest programming language to use on the VAX.

FORTRAN — A high level math-oriented programming language which is ideal for engineering applications. For this kind of application, FORTRAN is faster and more efficient than BASIC.

MACRO — This is an assembly language which uses VAX MACRO instructions as substitutes for strings of assembler instructions. It's closer to machine language than the other programming languages available, so it works faster and takes up less space in the computer, but it is more difficult and time consuming to use than the high level languages.

PASCAL — A high level structure programming language with more capabilities than FORTRAN, used

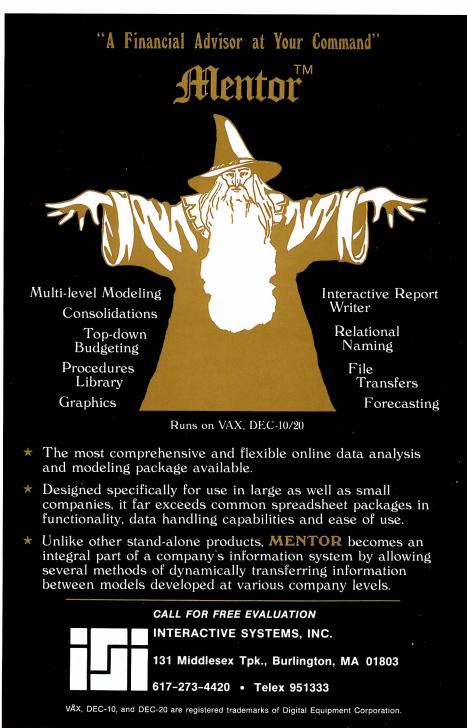
mainly for math and science applications.

DSR (Digital Standard Runoff) This is a text formatter which allows you to manipulate, format and justify text files into sequentially numbered pages. You can direct DSR to provide title pages, footnotes, tables of contents, indexes and appendices for text files which it processes. Through the use of EDT and DSR, each VAX user has extensive word processing capabilities at his or her disposal. DSR is not one of the most docuheavily mented features of VMS and takes practice and trialand-error experimentation on the part of a new user. However, by reading the VAX-11 DSR User's Guide and VAX/VMS Users Introduction and practicing with the various DSR commands and flags, you should soon be able to produce

pearing documents efficiently and rapidly.

UTILITIES — VMS provides utility programs which generate statistics on CPU and disk utilization, allow you to monitor system performance, provide a facility to back up files on the system to disk or magnetic tape, and allow you to convert files from one format to another. Of particular interest to users are the HELP, MAIL and PHONE utilities. HELP gives you on-line help on the use of the command language and system services and utilities, while MAIL and PHONE permit you to send messages to other users, or have a two-way conversation between different users.

THE VT-100 TERMINAL



CIRCLE 223 ON READER CARD

terminal consists of a black and white CRT display screen and housing and a separate typewriter-like keyboard with a numeric keypad. It's similar to a home computer in appearance and use. The VT-100 has more keys, a <RETURN> kev instead of an "enter" key and an on/off toggle switch at the lower left rear of the video display housing, but it is used in much the same fashion as a home computer.

The VT-100

Setting Terminal Characteristics

At the upper left hand corner of the VT-100 keyboard is a key labeled "SET-UP." This key is used to set the characteristics of the terminal, including the data transmission rate. screen brightness. normal or reverse video and audible nonaudible keyclick functions. If you are unable log in to

professional ap-

the computer system and you are sure that the computer cable and keyboard cable are plugged in properly, you should verify that the terminal is set to operate at the same speed as the computer system through the use of the SET-UP function. To do this, press the SET-UP key followed by the the numeric 5 key under the SET-UP A/B indicator light. This will produce the SET-UP B display which shows the transmit and receive speeds in the lower right hand corner of the screen. These speeds can be changed by depressing the numeric 7 and 8 keys while in the Set-Up mode. Once you have matched the terminal speed to the computer speed, press the SET-UP key again to exit the Set-Up mode. For more information on using the VT-100 in the Set-Up mode, refer to the VT-100 User Guide.

Although VAX systems are commonly set to communicate with video terminals at 4800 or 9600 baud, you should consult your System Manager to determine what speed is applicable on your system.

VT-100 Features

The VT-100 has several features that you should become familiar with. The first is the type-ahead buffer. This buffer is a memory within the VT-100 which permits you to type commands into the terminal while it is still processing an existing command. For instance, if you issued the DIRECTORY command and the terminal is listing the files in the directory, you can enter the command SHOW TIME as the files are being listed. As soon as the directory listing is complete, the command SHOW TIME and the system date and time will appear on the screen. The SHOW TIME command is stored in the type-ahead buffer and does not appear on the screen until the existing command has been processed. Using the type-ahead buffer, you can enter a series of commands or program responses as rapidly as you wish, even though the computer system may not yet be ready to act on them. Be careful when you do this, because data stored in the type-ahead buffer is displayed when it is input to the computer, not when you initially type it in at the terminal. You can flush data from the type-ahead buffer by entering < CTRL/X >, which will be discussed later in this section.

The <RETURN> key is used to input data or a command string to the computer after it has been typed at the terminal. It acts like a period, indicating the end of a command or data string. All commands and data that you type at the terminal must be followed by <RETURN> before the computer will accept the information.

You can use the < DELETE> key to rub out characters if you notice an error as you are typing a command. If you hold the < DELETE> key down, it will continue to erase characters on the current line from right to left. Make sure you use the < DELETE> key instead of the Backspace key for this purpose. The Backspace key will move the cursor from right to left through text, but it will not erase the characters that it moves over. This will result in errors when you depress the < RETURN> key and the character string is input to the computer.

Control Key Commands and Functions

Another useful feature of the terminal is the < CTRL>

or Control key. When the <CTRL> key is depressed and held down while specified alphabetic keys are depressed, certain functions are performed, or commands are issued or modified. Some of the most common <CTRL/letter> functions are listed and described here:

<CTRL/C> — Initiates the login sequence when you begin a terminal session, interrupts the command currently being processed or the program being executed.

Assume that you have directed the computer to display a long listing at your terminal. While the list is being displayed, you decide that you have seen enough and want to enter a new command immediately rather than look at the rest of the list. Type < CTRL/C> and the listing will halt and the \$ prompt will appear.

<CTRL/0> — Alternately suppresses and continues the display of output at the terminal.

Again, the computer is displaying a long list. Depressing < CTRL/O> once will cause the terminal to stop displaying additional lines of the listing. Unlike < CTRL/C> and < CTRL/Y>, < CTRL/O> will not cancel the list command — the computer continues to output the listing, but it is ignored by the terminal. If you type < CTRL/O> again, the listing will reappear. Like the on/off switch of your television set, < CTRL/O> affects what you see on the screen, but has no effect on what is being transmitted.

<CTRL/R> — Retypes the current input line and
repositions the cursor at the end of the retyped line.

This is handy if you have made mistakes in a command line and then corrected them. To see the corrected line as it will appear to the computer, press <CTRL/R> and the line will be retyped with no embedded blanks or correction characters. You can now enter the command by pressing <RETURN> or make additions or further changes to it prior to entering it into the system.

<CTRL/T> — Displays process statistics.

When you depress < CTRL/T>, you get an immediate one-line display of your process. This includes your terminal number, username, time, what you are running and the total CPU time that you have used since you logged in. The display also shows the current memory size of your process and the number of page faults and I/Os that your process has generated. You can use < CTRL/T> at any time after you have logged in without affecting your process. This permits you to obtain a status display when you are editing, or when you cannot access the DCL interpreter and \$ prompt.

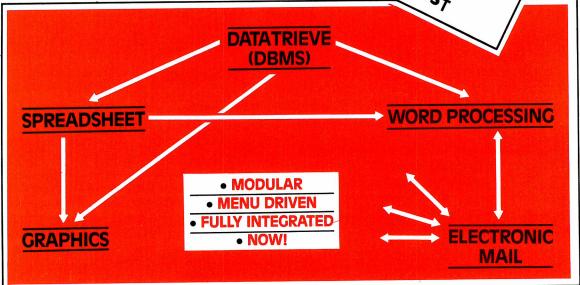
<CTRL/U> — Discards the current input line.

If you have made a mistake in a command or data line and prefer to erase it instead of correct it, or decide to enter a new command or no command at all, press < CTRL/U>. The information that you have typed is ignored by the system and the \$ prompt returns.



HAMILTON'S VOX LINKS WILL INTEGRATE YOUR EXISTING SOFTWARE INTO A COMPLETE OFFICE SYSTEM!





RENTAL OR SALE of VOX complete or by module with or without VAX hardware. **TIMESHARING** of VOX and full range of DEC layered products and software tools.

HARDWARE CONFIGURING All DEC systems and peripherals together with Hewlett Packard and Tektronix graphic devices, Dataproducts and letter quality printers, pretested and installed on your site with manufacturers' warranty and maintenance available.

VAX, Datatrieve are trademarks of Digital Equipment Corp./DRS, trademark of Advanced Data Mgt./Supercomp 20, trademark of Access Technology Inc./Digicalc, trademark of WHY Systems Inc./CT*OS, trademark of Compu-Tone Inc./Word 11, trademark of Data Processing Design Inc./D-PICT/B, trademark of Data Plotting Services, Inc./Fingraph, trademark of Fingraph Corp.

†Other packages being linked. Call us for details or requests.

HAMILION

THE HOUSE OF VAX

HGL Software

Hamilton Rentals

6 Pearl Court, Allendale, NJ 07401 TOLL FREE 800-631-0298 In New Jersey 201-327-1444 415 Horner Ave, Toronto, M8W4W3 **TOLL FREE Ont. & Que. 800-268-2106** All other prov. 800-268-0317 In Canada 416-251-1166

NEW YORK • DALLAS • MONTREAL • CALGARY • LONDON • PARIS • DUSSELDORF

<CTRL/X> — Flushes the type-ahead buffer.

If you have entered information in the type-ahead buffer and want this information erased and disregarded, press < CTRL/X> to empty the type-ahead buffer.

 $<\!$ CTRL/Y>- Interrupts command or program execution and returns control to the DCL command interpreter. The use of this key will abort an active command, program or utility and immediately return the DCL $\$ prompt. For this reason, be careful when using $<\!$ CTRL/Y> $\!$. You can disable this function if you desire to do so by entering the DCL command SET NOCONTROLY.

<CTRL/Z> — Indicates end of file for data entered from the terminal.

<CTRL/Z> is used like a period to indicate to the system that no more data is to be entered in a file. It is commonly used in programs and utilities.

LOGGING IN

To log in to the system, start by flipping the toggle switch located at the lower left rear of the VT-100 terminal up to the "on" position. In about thirty seconds, you should see a flashing bar of light at the upper left hand corner of the screen. This means that the VT-100 is ready to accept your input. If the cursor bar does not appear, check to see if the VT-100 is plugged in and make sure that the cable from the computer system and the keyboard cord are plugged in. To initiate your session with the computer, press < RETURN>, < CTRL/C> or < CTRL/Y> on the VT-100 keyboard. A site-specific welcome message such as the following should appear:

*** VAX/VMS V3.4 — POISE VAX-11/750 Roswell, N.M. USA *** followed by:

USERNAME:

At this point, type in your account name on the keyboard and press <RETURN>. Your account name is printed on the screen as you type it in. When you press <RETURN>, the system reads your account and name and displays the following prompt:

PASSWORD:

Now, type in your password. It is entered into the system, but it's not displayed or "echoed" on the screen. Thus, you don't have to worry about anyone "looking over your shoulder" as you log in. Press < RETURN > and the terminal will display another welcome message such as:

Welcome to POISE and VAX/VMS V3.4 s

or

USER AUTHORIZATION FAILURE

Once you are logged in to the system, the terminal screen will display the Digital Command Language (DCL) "\$" prompt character, which indicates that the system is ready to accept your commands. The "\$" prompt will appear after the execution of every command that you issue to the system — in short, it is your cue to continue your conversation with the VAX. However, if USER AUTHORIZATION FAILURE is displayed on the screen, you typed in the wrong password, misspelled your password, or tried to access another user's account. To try to log in again, press < RETURN>, < CTRL/C> or < CTRL/Y> to go back to the first welcome message and the USERNAME: prompt.

If you are still unable to log in and establish communications with the computer, your terminal may be set at a different character transmission rate (baud rate) than the computer is operating at. To change the baud rate of your terminal, see the Set-Up section of the VT-100 manual, or ask your System Manager or other designated individual for assistance.

LOGGING OUT

When you have finished a session with the VAX, you should always LOGOUT prior to shutting off your terminal. Until you do log out, VMS maintains an active process in your account and keeps a record of how much elapsed time and CPU time that your account has used, just like a taxi meter. In addition, if you shut off your terminal without logging out, another person can come along and turn the terminal on and get access to your account and files. Of course, any computer time that this person uses will be billed to your account. So, remember to type in:

\$ LOGOUT < RETURN>

or

\$ LO < RETURN >

before you shut your terminal off.

Within a few seconds of entering the LOGOUT command, you will see a display confirming that your account is logged out of the system. This display also shows accounting information including, among other data, the total elapsed time that the account was logged in and the total CPU time charged to the account during the terminal session. This information is stored in a special accounting data file that the System Manager uses to produce periodic reports that detail computer usage. The information is stored automatically, thus there is a permanent record of who is using the system, how long it has been used, and at what times the usage is taking place. This is not only a management tool, but also a deterrent to unauthorized system use. CPU time is expensive, and it's vital for the System Manager to have a method to make sure that outsiders or unauthorized individuals are not accessing the VAX and using its resources.

1.0 DIGITAL COMMAND LANGUAGE

In order to get a computer system to work for you, you must converse with it in a command language that it understands. Dialogue with a VAX system is carried on in Digital Command Language, or DCL. DCL consists of about 200 commands or keywords, each of which may have a number of modifiers or parameters and qualifiers. DCL is easy to learn and use because each command is a verb which causes the system to do almost exactly what it says. Thus, if you want to run a given program, you type in RUN programname and the program is executed for you. As in the English language, RUN is a verb and "programname" is an object.

Most DCL commands have parameters and qualifiers. You normally do not have to memorize these because of the extensive use of defaults and prompts in DCL. If you enter a DCL verb that requires a qualifier without entering the qualifier, the command language interpreter will prompt you for a qualifier rather than issue an error message. For example, the COPY command requires that you enter the

filename that you are copying from as well as the destination filename that you are copying to. If you don't specify these two parameters, you will receive the following prompt:

\$ COPY < RETURN >

- \$__From: (Here you enter your source file)
- \$__To: (The destination file is entered here)

You can also reduce the number of keystrokes that you must enter by using defaults. The **TYPE** command, which displays a file at your terminal, works with a default filetype of .LIS. If you have a file named SUSANJ.LIS that you wish to display, you could enter:

\$ TYPE < RETURN >

\$__File: SUSANJ < RETURN >

and SUSANJ.LIS would print at your terminal. Remember that the filetype must be .LIS for this default to work. If your file was named SUSANJ.TXT, you would have to specify SUSANJ.TXT, or the DCL interpreter would return an error message stating that it could not find the file SUSANJ.LIS that it looks for by default.

Digital Command Language and default values for specific commands are extensively documented by the VAX/VMS HELP facility and the VAX/VMS Command Language User's Guide. Refer to one of these sources for information on specific commands and their uses and default characteristics.

1.0 FILE STRUCTURE AND MANIPULATION

Certain conventions apply to the naming, manipulation and usage of files on the VAX system. Some of these conventions are presented and briefly discussed here. Detailed information on files is available in several of the Digital documentation manuals.

A file is a group of related records which is stored on magnetic tape or a disk pack. On a VAX/VMS system, each file must have a unique name (filename) and other attributes which identify it to both the system and the user. The complete name for any given file is called its file specification. Rules for assigning filenames and file specifications follow the sample file specification and explanation below:

A complete file specification consists of the device name and directory name where the file resides, followed by the filename, filetype and version number of the file:

DRAO: [POISE] SUSANJ. TXT; 222

^	^	^	^	^	
:	:	:	:	:	Version Number
:	:	:	:		
:	:	:	:		Filetype
:	:	:			
:	:	:			Filename
:	:				
:	:				Directory name
:					
:.					Device name

In this example, version number one of a text file called SUSANJ is located in the directory [POISE] on disk DRAO. If DRAO:[POISE] is your default disk and directory, you can omit the disk and directory from your file specification when you reference the file. For example, to display the file at your terminal, you would enter:

\$ TYPE SUSANJ.TXT;222

and the contents of SUSANJ.TXT;222 would print at your terminal. If version number 222 of this file were the only version of SUSANJ.TXT in your directory, or if it were the highest version number of the file, you could omit the version number from your TYPE command. By default, the highest version number of a file is the one that the DCL interpreter will act upon unless you explicitly specify otherwise.

The following rules apply to file specifications on the VAX:

- 1. Punctuation is required to separate the device name (:), directory name ([. . .]), filename and filetype (.) and version number (;) from each other.
- 2. No blank spaces are allowed within any portion of a file specification.
- 3. You may use any alphanumeric names that you wish within a file specification, but the first character of each portion except the version number must be alphabetic. The version number must consist of octal (base 8) numbers only.
- 4. Directory names and filenames may consist of up to nine characters. Filetypes are limited to three characters, and version numbers are limited to five digits.

1.1 Filetypes

Certain filetypes have default values which are known to the system. A complete table of default file types may be found in the VAX/VMS Command Language User's Guide. A list of some of the more common default filetypes that you may see in your directory is presented here:

- 1. BAS VAX BASIC source program.
- 2. **COM** A DCL command procedure.
- 3. DAT A data file used by a program.
- 4. DIR A directory file.
- 5. **DIS** A distribution file, as used by MAIL.
- 6. **EXE** Denotes an executable program.
- 7. FOR FORTRAN source program.
- 8. **EDT** An EDT initiator file.
- 9. ${\it LIS}$ Indicates a listing file. This is the default file type for the TYPE and PRINT commands.
 - 10. MAI A MAIL message file.
 - 11. MAR MACRO-11 source program.
 - 12. **MEM** A formatted text file generated by DSR.
 - 13. **OBJ** An object program output by a compiler.
 - 14. RNO A text file to be input to DSR.
 - 15. **SYS** A system image file.
 - 16. **TMP** A temporary file.
 - 17. TXT A text file or MAIL command output file.

1.2 Wildcarding

VMS supports the use of "wildcards" in file manipulation. Two characters are valid for wildcarding — the asterisk and the percent sign. They work as follows:

Assume that you have a number of versions of the file SUSANJ.TXT and you wish to print all of them out. Instead of entering:

\$ PRINT SUSANJ.TXT;(version-number)

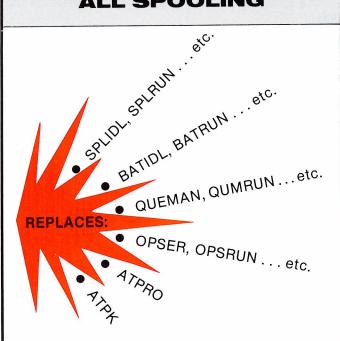
for each version of the file, you could type:

\$ PRINT SUSANJ.TXT;*
and all the versions of SUSANJ.TXT in your directory would

VERSION 2.2 NOW AVAILABLE

QUE.11 — V2.2

ONE JOB SPOOLER FOR RSTS/E CONTROLS ALL SPOOLING



QUE.11:

- DEC QUE Compatible
- Block letters on spooled header page
- One job controls all spooling
- Saves small buffers and job slots
- Spawns jobs as needed
- Handles line printer and keyboard spooling
- Controls as many BATCH JOBS as pseudo-keyboards
- Full parameter replacement in QUE
- calls "DO" command replaces indirect processors
- QUEMAN SYS call supported
- Program deliveries NOW
- Only \$1500 single CPU license
- Trial Version \$100

For more information contact:

On Track Systems, Inc.

P.O. Box 245 Ambler, PA 19002-0245 Phone: 215/542-7133 In Europe:

Procyon Informatics, Ltd. 19 St Kevins Road Dublin 8, Ireland

CIRCLE 11 ON READER CARD

RESOURCE ACCOUNTING UNDER VAX/VMS

. . . continued from page 47

Each of the eight accounting records consists of a record header that defines the record type followed by a variable number of information packets. The number and type of information packets depends on the accounting record type, which is defined in the record header. The record header is 12 bytes long. The bytes are defined as follows:

Symbolic offset	Byte Offset	Length	Comments
ACR\$W_TYPE	0	2	Contains 5 sub fields
ACR\$W_LENGTH	2	2	Length of accounting record
ACR\$Q_SYSTIME	4	8	System time

Let's now consider the bit substructure of the type field. The type field always comprises the first two bytes of the accounting record:

	Symbolic offset	Bit Offset	Length	Comments
AC	R\$VPACKET	0	1	Packet header flag
AC	R\$V_TYPE	1	7	Record type
AC	R\$V_SUBTYPE	8	4	Record subtype
AC	R\$V_VERSION	12	3	VMS accounting record version
AC	R\$V_CUSTOMER	15	1	Customer record flag

The packet header flag field is a one bit field which indicates whether this is a record header or an information packet. The bit is clear for record headers and set for information packets.

The type field is a seven bit field that indicates which of the eight record types the accounting record is. The following chart indicates the possibilities:

Symbolic constant	Value	Record type
ACR\$K_PRCDEL ACR\$K_IMGDEL ACR\$K_SYSINIT ACR\$K_LOGFAIL	1 3 5 7	Process deletion record Image deletion record System initialization record Login failure record
ACR\$K_PRINT	8	Print job termination record
ACR\$K_USER	9	User defined record
ACR\$KFILEFL ACR\$KFILEBL	13 14	Accounting file forward link record Accounting file backward link record

The subtype field is a 4 bit field that is used only for process deletion and image deletion record types. It indicates the process type with which the record is associated. The following chart indicates the possible values:

Symbolic constant	Value	Record type
ACR\$K_INTERACTIVE ACR\$K_SUBPROCESS ACR\$K_DETACHED ACR\$K_BATCH ACR\$K_NETWORK	1 2 3 4 5	Interactive process Subprocess Detached process Batch job Network process



10th ANNIWER SPECIAL OFF

The logical switch to color has never cost less. Now you can get immediate delivery on an Intecolor 2405D single evaluation unit at the special 10th Anniversary price of \$995 (U.S. domestic only

You'll have the advantage of vector graphics on an 80 column by 24 line screen, without sacrificing the most important capabilities you want from DEC's VT100/131 or VT220 terminals. Plus, the 2405D's eight colors convey more information, more quickly and with greater comprehension than VT100/131's monochrome

ANSI X3.64 compatibility. The 2405D is the ideal replacement or add-on terminal. Highly compatible with VT100/131/220 and numerous other ANSI X3.64 terminals, the 2405D is easily integrated into any ANSI X3.64 environment. It includes ASCII codes and a VT52 mode. The 2405D is also compatible with Intecolor's 2427 bit-mapped color graphics terminal.

With all the features you need. Terminal based vector graphics. Data transmission baud rates from

Our ANSI X3.64 terminal has all the VT100/131 and VT220 features you'll ever need in a conversational terminal. Plus eight colors and vector graphics. Available now.

50 to 19,200. English language menu set-up mode. Non-volatile set-up memory. Two full pages of screen RAM. In-line CRT. Auto degaussing. Powerful, 6MHz 8085 CPU with four hardware interrupts. Detached keyboard. Our 10th Anniversary \$995 offer includes 12 function keys (with 36 programmable functions), normally optional. Another 12 keys are optionally available for \$100. Shipment less than 30 days after acceptance of order.

Plus Intecolor Quality. Since 1973. Intecolor has been a pioneer in high quality color graphics terminals for industrial/scientific markets. And we are continually setting new stan-

dards of reliability.

10th Anniversary price good through 1/31/84. After that, our

regular single-piece price of \$1295 (100 piece: \$1075) goes into effect, and the first 12 function keys are \$95 with a total cost of \$1390. Save \$395 by acting now! Take full advantage of the color, vector graphics and ANSI X3.64 compatibility of the Intecolor 2405D at this special price.

For the name of the distributor or sales representative in your area, or for complete specs, ask about our 10th Anniversary \$995 special: Call 1-800-241-7595.



AN INTELLIGENT SYSTEMS COMPANY

Intecolor Drive, 225 Technology Park, Norcross, GA 30092, TWX 810-766-1581

C B E D I T
The \$200 RSTS/E*
W O R D P R O C E S S O R

Version 3.0 includes list processing, optional print-time operator input, on-screen help and directory management plus total device independence. All files are standard ASCII with no control codes. Data loss protection on system crash or operator error.

Window edit includes add, insert, global locate, global change, block replace, block delete, block copy, block move, document merge, etc. Fully formatted output with page numbers, headers, footers, margins, indentation, justified text, underscore, super/subscript, center, multiple spacing, etc. Automatic or manual pagination with unlimited document length.

Table driven Video and Printer modules support any VDTs (including 132 column and 66 line) and any printers. Mixed units on same system are OK.

Price: \$200 on 9-track/800 bpi plus shipping. Other media available. User's manual and Basic-Plus* source are included.

T. F. Hudgins & Associates, Inc. P.O. Box 10946 Houston TX 77292 Wes Seago 713/682-3651

*TM Digital Equipment Corporation

CIRCLE 9 ON READER CARD

Announcing MPR

A VERSATILE MACRO-LANGUAGE UTILITY FOR RSTS/E

Flexible enough for OEM's and End-Users

Functional highlights include:

- · Customized code generation.
- Nestable macros and "INCLUDE" statements with arguments.
- · Algebraic expression evaluation.
- Control file generation.
- Extensive debugging mode.
- Easy to use syntax.
- Complete documentation with sample macros.

Benefits:

- · Reduce software maintenance costs.
- Increase software portability.
- · Simplify operations and training.
- Compatible with your existing programming standards

Interested in significantly improving programming productivity?

CALL OR WRITE FOR DETAILED INFORMATION

- Noah Dixon -

. Star Plan Data Processing, Inc. .

5990 N. Green Bay Avenue Glendale, Wisconsin 53209 - (414) 228-1280

· Soon to be released for RSX and VMS

CIRCLE 178 ON READER CARD

The version field is a three bit field that indicates the version of VMS that created the accounting record. The following chart indicates the possibilities:

Symbolic constant	Value	Record type
ACR\$K_VERSION2	0	VMS version 2 record
ACR\$K_VERSION3T	1	VMS version 3 field test record
ACR\$K_VERSION3	2	VMS version 3 record

The customer flag field is a one bit field which indicates whether the record was written by Digital software or by customer software. The bit is clear for Digital software and set for customer software. User defined records written via the system service \$SNDACC leave this bit clear.

After the accounting record header comes one or more information packets. The number and type of information packets depends on the accounting record type, defined in the record header. There are currently six different information packets defined. The following is a chart listing the possibilities:

Symbolic constant	Value	Packet type
ACR\$K_ID	1	Process identification packet
ACR\$K_RESOURCE	2	Resource usage packet
ACR\$K_IMAGENAME	3	Image name packet
ACR\$KFILENAME	4	File name linkage packet
ACR\$K_USER_DATA	5	User defined data packet
ACR\$K_PRINT	8	Print job resource usage packet

This next chart shows the information packets used by each of the eight accounting record types. This chart indicates the "macro" record structure which follows the record header for each of the accounting record types. For brevity I will use the symbolic names:

Record type	Packet type(s)
ACR\$K_PRCDEL	ACR\$K_ID, ACR\$K_RESOURCE
ACR\$K_IMGDEL	ACR\$K_ID, ACR\$K_RESOURCE, ACR\$K_IMAGENAME
ACR\$K_SYSINIT	ACR\$K_ID, ACR\$K_RESOURCE
ACR\$K_LOGFAIL	ACR\$K_ID, ACR\$K_RESOURCE
ACR\$K_PRINT	ACR\$K_ID, ACR\$K_PRINT
ACR\$K_USER	ACR\$K_ID, ACR\$K_USER_DATA
ACR\$KFILEFL	ACR\$KFILENAME
ACR\$K_FILE_BL	ACR\$KFILENAME

Now let's consider the structure of each of the six information packets. Each packet begins with a packet header that comprises four bytes. The packet header structure models the record header structure stored in the first four bytes of the accounting record. The packet header structure is as follows:

Symbolic offset	Byte Offset	Length	Comments
ACR\$W_TYPE	0	2	Contains 5 sub fields
ACR\$W_LENGTH	2	2	Length of information packet

The bit substructure of the type field is identical to the type field in the record header. The only differences between the record header and the packet header type fields are that the packet flag bit (ACR\$V_PACKET) is set and the type field refers to the packet type rather than the record type. The subtype field is always zero and the customer flag is always clear, even for user data packets.

Now that we have the first four bytes of each information packet out of the way, we can begin to consider the data structure of each packet from byte five to the end. All packets except for the resource packet and the print packet are variable length. First, however, we must consider the field types stored in packets.

There are three basic types of data fields stored in packets. They are fixed length fields, offset fields or variable length fields. Up until now, all the fields we have considered have been fixed length fields.

Fixed length fields can be any of a variety of fixed length data types (e.g. byte, word, longword, quadword). The value of the field is the data we are ultimately interested in.

Offset fields are word fields that indicate where in the data structure the actual data begins. Offset fields introduce one level of indirection. We first have to go to the offset field and read its value and then use the value to index to the actual location of the data.

The third type of field is a variable length field. Variable length fields are always represented by counted strings. A counted string is a data structure where the first byte indicates the number of characters that are contained in the rest of the string. The rest of the string, which is the data portion, immediately follows the count byte. Counted strings can have from zero to 255 characters. Variable length data always follows all of the fixed length data fields and offset fields in all information packets.

Before I actually list the format of each of the information packets, please note. Within the data packets, not all fields always have meaningful data. For example, the owner process field in the process identification packet has meaning only if the record is a subprocess record. If a field has no meaning and is in the fixed portion of the packet, its value is zero. If the field is in the variable portion of the packet (i.e., it is a counted string), no space is allocated for it and the offset to the field is zero.

The following two abbreviations are used in the charts:

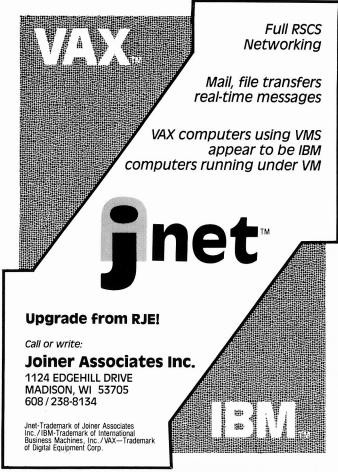
CAS = counted ASCII string

Var = variable position, depends on length of preceding variable data

Here are the formats of the six information packets:

ACR\$K_ID, process identification packet, variable length

Symbolic offset	Offset	Comments
ACR\$W_TYPE	0	Contains 5 sub fields
ACR\$W_LENGTH	2	Length of packet
ACR\$L_PID	4	Process ID of process
ACR\$L_OWNER	8	Process ID of owner (if subprocess)
ACR\$L_UIC	12	UIC of process (see footnote)
ACR\$Q_PRIV	16	Process privilege quadword



CIRCLE 222 ON READER CARD

PDP-11 VAX

Converting from RSTS/E to VAX?

Call the people who pioneered the techniques used to migrate from a RSTS/E environment to a native mode VAX/VMS environment with full access to all VMS services.

- Conversion Techniques & Aids Basic + to VAX-11 Basic BP2 to VAX-11 Basic
- Transportable Software Programming Techniques
- RSTS/E Block I/O from VAX-11 Basic

Applications not performing well on your VAX?

Applications designed for a PDP-11 environment may not perform as well as expected in a VAX/VMS environment since they do not use the full capabilities of VMS. Modular improvements designed to use capabilities unique to VMS can give you the performance you thought you were getting when you moved to a VAX. Call for information on:

- Application and System Software Design
- VMS Performance Tuning
- VAX-11 Macro Programming

DAVE FROBLE ENTERPRISES, INC.

116 Jennings Avenue Scottdale, Pa. 15683 (412) 887-7010

PDP-11, VAX, VMS, RSTS/E, Basic+, BP2, and VAX-11 Basic are trademarks of Digital Equipment Corporation.

ACR\$B_PRI	24	Base priority of process
ACR\$W_USERNAME	26	Offset to process username CAS
ACR\$W_ACCOUNT	28	Offset to process account CAS
ACR\$W_NODENAME	30	Offset to remote nodename CAS
ACR\$W_TERMINAL	32	Offset to process terminal CAS
ACR\$W_JOBNAME	34	Offset to job name CAS
ACR\$L_JOBID	36	Identification of job (if batch or print job)
ACR\$W_QUEUE	40	Offset to queue name CAS
ACR\$W_NODEADDR	42	Node address of remote nodename
ACR\$W_REMOTEID	44	Offset to remote process ID CAS
	46	Username CAS
	Var	Account CAS
	Var	Remote nodename CAS (if network job)
	Var	Process terminal CAS
	Var	Job name CAS (if batch or print job)
	Var	Job queue name CAS (if batch or print job)
	Var	Remote process ID (if network job)

Footnote: the UIC can also be addressed as follows;

ACR\$WMEM	12	UIC member of process
ACR\$WGRP	14	UIC group of process

The remote process ID is DECNET dependent. For example, if the connect initiate originated from another VAX/VMS node, the remote process ID is equal to the process ID of the remote process sending the connect initiate. If the connect initiate comes from a RSTS/E node, it is equal to the name that was declared in the message receiver table on RSTS/E.



CIRCLE 72 ON READER CARD

ACR\$K_RESOURCE, resource utilization packet, fixed length, 52 bytes

Symbolic offset	Byte Offset	Comments
ACR\$W_TYPE	0	Contains 5 sub fields
ACR\$W_LENGTH	2	Length of packet
ACR\$Q_LOGIN	4	Start time for process or image
ACR\$L_STATUS	12	Final exit status of process or image
ACR\$L_IMGCNT	16	Number of images executed (see footnote)
ACR\$L_CPUTIME	20	Total CPU time for process or image
ACR\$L_FAULTS	24	Page fault count for process or image
ACR\$L_FAULTIO	28	Page fault I/O count for process or image
ACR\$L_WSPEAK	32	Peak working set size of process or image
ACR\$L_PAGEFL	36	Peak page file usage of process or image
ACR\$L_DIOCNT	40	Direct I/O count for process or image
ACR\$L_BIOCNT	44	Buffered I/O count for process or image
ACR\$L_VOLUMES	48	Volumes mounted by process or image

Footnote: For process termination records, the image count field contains the number of images executed. For image termination records, the image count field contains the image number in the image execution sequence.

ACR\$K_IMAGENAME, image name packet, variable length

Symbolic offset	Byte Offset	Comments		
ACR\$W_TYPE	0	Contains 5 sub fields		
ACR\$WLENGTH	2	Length of packet		
ACR\$T_IMAGENAME	4	Image file name CAS		

ACR\$K__FILENAME, file name packet, variable length

Symbolic offset	Byte Offset	Comments
ACR\$W_TYPE	0	Contains 5 sub fields
ACR\$W_LENGTH	2	Length of packet
ACR\$T_FILENAME	4	File name CAS

ACR\$K_USER_DATA, user defined data packet, variable length

Symbolic offset	Byte Offset	Comments
ACR\$W_TYPE ACR\$W_LENGTH ACR\$T_USER_DATA	0 2 4	Contains 5 sub fields Length of packet User defined data CAS

ACR\$K_PRINT, print job utilization packet, fixed length, 40 bytes

Symbolic offset	Byte Offset	Comments
ACR\$WTYPE	0	Contains 5 sub fields
ACR\$W_LENGTH	2	Length of packet
ACR\$L_PRINTST	4	Final exit status of print job
ACR\$Q_QUETIME	8	Time print job was queued
ACR\$Q_BEGTIME	16	Time print job was started
ACR\$L_SYMCPUTIM	24	Symbiont total CPU time for print job
ACR\$L_PAGECNT	28	Total number of pages printed for print job
ACR\$L_QIOCNT	32	Symbiont QIO issued count for print job
ACR\$L_GETCNT	36	Symbiont RMS GET count for print job

Part 2 continues next issue.

Introduction to the VAX/VMS...continued from page 59

be printed on the line printer. If your directory also contained the files SUSANJ.DAT and SUSANJ.COM and you wanted to display the contents of all the versions of all the files with the file name SUSANJ at your terminal, you could enter:

\$ TYPE SUSANJ. *:*

to obtain this information. If you wanted to print the entire contents of your directory on the line printer, the command:

\$ PRINT *.*;*

would do the job for you. Because the wildcard can be used in any portion of filename.filetype;version specification, the following commands could also be used:

- \$ PRINT *.TXT;* (Print all versions of all .TXT files)
- \$ PRINT *.*;1 (Print all files with a version number of 1)
- \$ PRINT S*.*; (Print all versions of all files starting with "S")
 The "%" wildcard character allows you to specify all
 that contain any single character in a given position

files that contain any single character in a given position. The "%" identifies the position that is wildcarded. In the example above, PRINT S*.*;*, every file beginning with the letter S would be printed, regardless of the letters and number of letters following the S. By using the "%" wildcard, you can be more selective. Assume that in addition to the SUSANJ files, you also have the files SUSANJ1.TXT through SUSANJ13.TXT in your directory. To print the files SUSANJ1.TXT through SUSANJ9.TXT (remember that while version numbers on the VAX are represented in octal numbers, any decimal number is valid within the filename portion of a file specification) on the line printer, enter the command:

\$ PRINT SUSANJ%.TXT;*

This will print the desired files, but will **not** print SUSANJ10.TXT through SUSANJ13.TXT **or** any of the SUSANJ.TXT files, as the " * " wildcard character would.

After you gain some experience with the system, you will find that wildcarding can be very beneficial and will save you time and keystrokes when you interact with the computer system.

1.3 File Manipulation

Now that you know how to use file specifications and how to reference particular files or groups of files, you should learn some of the commands and conventions that are used to move, modify or delete files. In the following examples, assume that you have a directory containing two files, SUSANJ1.TXT;1 and SUSANJ2.TXT;1.

First, make a copy of SUSANJ1.TXT;1:

\$ COPY SUSANJ1.TXT *.* < RETURN>

The directory will now show:

SUSANJ1.TXT:1

SUSANJ1.TXT;2

SUSANJ2.TXT;1

If these text files were chapters of a document, you might want to put them together sequentially in a single text file. The DCL APPEND command allows you to do this:

\$ APPEND SUSANJ2.TXT SUSANJ1.TXT < RETURN> The directory now shows:

SUSANJ1.TXT;1

SUSANJ1.TXT;2

SUSANJ1.TXT;3

SUSANJ2.TXT;1

This is because SUSANJ2.TXT:1 was appended to the end of SUSANJ1.TXT:2. As the APPEND command actually creates a new file from two existing files, the new file is assigned a new version number. Note that even though SUSANJ2.TXT was appended to SUSANJ1.TXT, a copy of SUSANJ2.TXT still exists in your directory.

Now you have three files with the same filename and filetype in your directory. If the first two versions of SUSANJ1.TXT are obsolete, you can remove them by entering:

\$ PURGE SUSANJ1.TXT < RETURN>

And you can remove SUSANJ2.TXT by typing:

\$ DELETE SUSANJ2.TXT;1 < RETURN>

or

\$ DELETE SUSANJ2.*:* < RETURN >

The use of wildcards is optional, but it's easier to type in ""; "" than the filetype and version number.

Your directory will now show only one entry: SUSANJ.TXT;3

If you wanted to input this file to DSR under a different name and with the default Runoff filetype, .RNO, you could use the **RENAME** command:

\$ RENAME SUSANJ1.TXT SUE.RNO < RETURN >

and the directory would show:

SUE.RNO; 1

The output of DSR is a file with a filetype of .MEM. If you no longer needed the .RNO file, you would enter:

\$ DELETE SUE.RNO;1 < RETURN>

\$ DELETE SUE.*;* < RETURN>

and you would be left with an empty directory.

For further information and examples of file manipulation with Digital Command Language, refer to the VAX/VMS Command Language User's Guide.

1.2 SYMBOLS AND COMMAND PROCEDURES

Through the use of symbols and command procedures, you can interact with your VAX/VMS system more easily and rapidly. Symbols and command procedures help you reduce the number of keystrokes and individual DCL commands needed to get your system to do what you want it to.

Symbols

A SYMBOL is a form of "shorthand" that you can use to assign a short name to a DCL command string. By creating symbols, you can reduce a forty character DCL command string to a synonym that is just a character or two long. Here's an example of symbol assignment:

Assume that you have second level subdirectory that you use frequently. In this example, we will use [POISE] as the root directory, [.EDITOR] as the first level subdirectory, and [.NOTES] as the second level subdirectory. Using normal DCL conventions, every time you want to enter this subdirectory, you must type in:

\$ SET DEFAULT [POISE.EDITOR.NOTES]

before you can access the files in [.NOTES]. By using the symbol assignment string of ":==" and a symbol name of WR*ITE, you can equate the symbol WR*ITE to the SET DEFAULT command string. The asterisk within the symbol

indicates that all letters to the right of the asterisk are optional — you can type the entire symbol WRITE or simply the first two characters, WR, and the DCL interpreter will issue the **SET DEFAULT** command string. To assign and use your symbol, enter the following:

\$ WR*ITE: = SET DEFAULT [POISE.EDITOR.NOTES]
Now, whenever you want to access the [.NOTES] subdirectory, all you have to type in is "WR".

You can see that this is a timesaving feature, particularly if you have a large number of subdirectories or if you use the same DCL command strings frequently. You can equate a different symbol to each of these command strings just as was done above, place the symbols in your login.com procedure, and save time and keystrokes during your terminal sessions with the computer system.

Command Procedures

You can also write COMMAND PROCEDURES to make your interaction with the computer faster and easier. A command procedure is a file of DCL command strings which will be executed sequentially when the procedure is invoked by typing in the execute command, "@", followed by the name that you have given the procedure.

Command procedures can be simple sequences of DCL commands that are executed in the order that they appear, or they can be sophisticated "programs" that act according to parameters that you establish, or predefined conditions that exist within the computer system. You can set up a procedure that will do different things based on your response to questions that you have established in the procedure, or you can write the procedure to work differently based on the time of day, based on the system parameter of time. An example of this is a procedure that would display "Good Morning" before twelve noon, "Good Afternoon" until six P.M. and "Working Late Again?" between six P.M. and midnight.

To write a command procedure, you use an editor such as EDT or you use the DCL CREATE command to build the file. In either case, the following rules must be observed:

- 1. A command procedure must be named Procedurenam.COM. The filetype .COM identifies the file to the DCL interpreter as a command procedure.
- 2. Each subsequent line must start with a \$. The exception to this rule is data elements, such as a "YES" or "NO" response. These are entered verbatim, without the DCL \$ prompt.
- 3. Comments or descriptions which are not intended to be executed must be preceded by \$!. The exclamation point causes everything to the right of it to be ignored by the DCL interpreter.
- 4. The procedure is invoked by typing @PROCEDURE-NAME. If you assign a filetype other than .COM to the procedure, you must enter a period and the three character filetype after the procedure name.

In a complicated command procedure, it is often a good idea to provide a way for the system to get out of the procedure if something goes wrong. This can be done by typing:

\$ ON ERROR GOTO EXIT

as the first line of the procedure after the title, and:

\$ EXIT:

as the last line of the procedure.

In addition, be sure you test the command procedure thoroughly. You can do this by entering the DCL command SET VERIFY and invoking the procedure, and entering SET NOVERIFY at the conclusion of the procedure. By doing this, every line of the procedure will be displayed on the terminal screen as it is executed. This lets you see what the procedure is doing as it executes, and allows you to debug or correct the procedure as needed. This is especially important if you are creating or modifying a login.com procedure. If you make a serious error in your login.com and fail to test it before you log out, you may be unpleasantly surprised the next time you try to log in to your account.

A sample LOGIN.COM procedure is provided below. It displays a greeting, the system time and assigns symbols for commonly used commands. It is invoked when you log in to the system, or whenever you type "@LOGIN". You may wish to tailor it to your site and use it in as the login.com for your own account. For complete information on symbols and command procedures, refer to the VAX/VMS Users Introduction, VAX/VMS Primer and the VAX/VMS Guide to Using Command Procedures.

```
$ SET NOON
$ WRITE SYS$OUTPUT "
                          SAMPLE LOGIN.COM PROCEDURE
$ WRITE SYS$OUTPUT "
                           (insert your username here)
$ WRITE SYS$OUTPUT "
$ WRITE SYS$OUTPUT "
$ WRITE SYS$OUTPUT " "
$ SHOW TIME
$ !
$ !assign useful symbols
$!
$ DI*RECTORY:== DIRECTORY/SIZE/DATE
$ EDT:== EDIT/EDT
$ FR*EE: == SHOW DEVICE DRAO:
$ PA*TH:== SET DEFAULT [root-directory]
$ PU*RGE:== PURGE/LOG
$ PRO*CESS:== SHOW PROCESS/ALL
$ QU#TA:== SHOW QUOTA
$ SQB:== SHOW QUEUE SYS$BATCH
$ SQP:== SHOW QUEUE SYS$PRINT
$ SD:== SET DEFAULT
$ ST*ATUS:== SHOW STATUS
$ TI*ME:== SHOW TIME
$ WHE*RE:== SHOW DEFAULT
$ W80:== SET TERM /WIDTH=80
$ W132:== SET TERM /WIDTH=132
$ ! system environment symbols
$ !
$ MEM*ORY: == SHOW MEMORY
$ SDA:== SHOW DEVICES/FULL/ALL
$ SYS:== SHOW SYSTEM
$ WHO: == SHOW USERS
$ WHE*RE:== SHOW DEFAULT
$ EXIT
$BATCH:
$ SET VERIFY
$ EXIT
```

We will continue to bring you excerpts from Mr. Shannon's valuable guide in February. Be watching for "Your Account," "Account Security," "The HELP Command," "Mail Utility," "Phone Utility," and more . . .

The information in this document is subject to change without notice and should not be construed as a commitment by the author. The author assumes no responsibility for any errors that may appear in this document. Any opinions expressed in this document are solely those of the author.

The following are trademarks of the Digital Equipment Corporation: DEC, DECUS,

PDP, VAX, VMS, VT.

UNITRONIX represents DIGITAL

PDP 11/23
SYSTEMS
SYSTEMS
With 256KB*
With 266KB*
With 266KB*
With 266KB*
With 266KB*
With 266KB*
With 266KB*
With 276KB*
With

TERMINALS

VT100 LA12 VT101 LA34 VT102 LA50 VT125 LA100 VT131 LA120

CPUs

PDP 11/23, 11/23+, 11/24, 11/44 & VAX

PCs

INTERFACES MODEMS

COMPLETE DATA SYSTEMS

with Unitronixdeveloped standard or customized applications software.

VT, LA, LSI, PDP, VAX are trademarks of Digital Equipment Corporation



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, accessories 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, inhouse programming, customer service, sales and marketing, component testing and complete warehouse facilities.



CORPORATION

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



CALL FOR LATEST PRICES

(201) 231-9400

KIM. PROGRAM TO TRANSFER FILES TO AND FROM TU58 CARTRIDGE TAPES

By Michael Mowat, Marine Laboratory, Aberdeen, Scotland

Many users of RSTS systems have bought TU58 Dectape 2 cartridge drives only to find that reliable operation is not possible when time sharing under RSTS. Although direct access to this drive is in general not satisfactory, it is possible to use the device if DEVICE HUNG errors are trapped (although the errors will still be logged). In particular the transferring of disk files to and from the TU58 can be handled by a program called KIM which is similar in function to the system utility FIT. This program can handle the TU58 either non file structured or as an RT11 file structured device. The program has been extended to store RSTS/E file attributes and even RMS file attributes if required. The program can also list a TU58 file to the user's terminal. Transfers to or from other non file structured devices or from devices with record sizes other than 512 bytes are not possible. The device is, however, rather slow. If the device is configured at 9600 baud, an effective baud rate of about 2000 can be achieved.

Description of the program KIM

There are four formats possible for the TU58 tape.

- 1. Non file structured. The data is terminated by a
- 2. RT11 format. Completely compatible with RT11 systems, file structured.
- 3. RSTS/E format. This is really RT11 format but the extra bytes in the directory are used to store the account number, protection codes, etc. and the device will behave as if it were a RSTS/E device.
- 4. Extended RSTS/E format. This is the same as RSTS/E format except that file attributes are also stored and recovered.

RT11 format should be used when communicating with RT11 systems. RSTS/E format tapes will be readable by such systems but there will be no distinction between two files of the same name on different accounts.

Restrictions

Version 1 of KIM will handle only one dual TU58. It should be possible to modify the program if more are needed. Due to hardware limitations only one user can gain access to each unit at a time. It is considered not worth trying to circumvent this restriction, as the seek time involved in accessing two data sets independently and concurrently would be excessive. Files are therefore accessed only sequentially.

Initializing a TU58 tape

This may be done only on a privileged account. Start the

program by RUN &KIM and type one of the following commands.

DDn:/ZE for RSTS/E format
DDn:/ZE/RT11 for RT11 format
DDn:/ZE+ for extended RSTS/E format
Non file structured tape need not be initialized but may only be written by a privileged user.

Non standard directory sizes

The number of directory segments allocated is normally four except for the extended RSTS/E format where eight segments are allocated. Other numbers in the range n=1 to 31 can be used by using the switch /ZE:n or /ZE +:n. Larger numbers may be required on tapes with many small files especially if a lot of deleting and rewriting is contemplated.

Copying files

The command for copying files is similar to that used by PIP except that the switches are different. It is in the form

device:[n,m]file.ext = device:[p,q]file.ext either the input or output device or both may be DDx: the default for device: is SY:, x may be 0 or 1, if omitted the default is 0. Both the account and filename may contain wildcard specifications. If the account is omitted the current account is assumed. If the output name is omitted or is * . * the actual input name is used. The following are valid copy commands

DD:[34,23]FRED.BAS = [23,12]JIG.BAS DD: = FRED.BAS *.* = DD:FRED.* DD1: = DD0:*.BAS

The no supercede switch

If /NS is added to the file copy command the file will not be copied if it already exists on the output device.

The detach switch

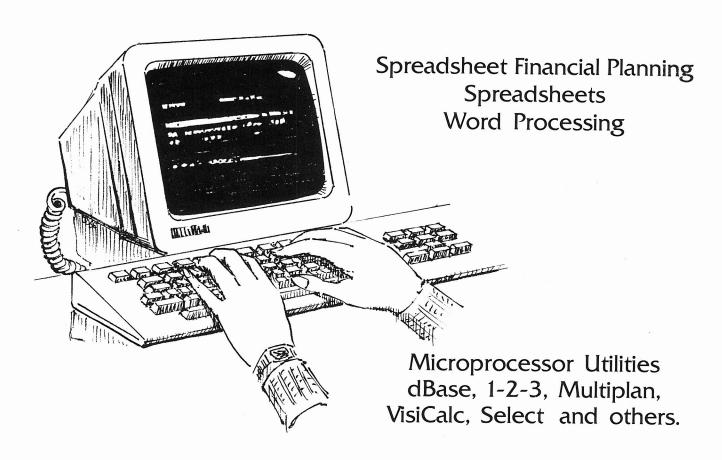
Because the TU58 devices are slow the job may be detached when copying large or many files by using the /DT switch. When the job is finished or if there is an error, a message will be sent to the initiating terminal.

Indirect command files

If a number of operations is to be carried out the commands can be put into indirect command files. These can be nested three deep. The command to call an indirect com-

TRAINING

DECmate II Rainbow 100 Professional 350



Classes offered in Philadelphia or your location. Taught by experienced professionals.

ON TRACK SYSTEMS, INC.
921 Bethlehem Pike, Springhouse, PA 19477 • 215-542-7008

mand file is @filename (default is .CMD). The /DT switch may be used; e.g.

RUN &KIM @FRED/DT

Non file structured TU58

Only three command formats may be used as follows device:[ppn]filename = DD:/NFS

DDx: = device:[ppn]filename/NFS

DDx: = DDy: /NFS

Output to TU58 non file structured is privileged. The last format should not be used to copy complete file structured tapes as the directory may be corrupted. Non file structured input is terminated on reading a blank block. Data tapes produced on other devices should not contain embedded blank records. A non file structured tape which contains blank blocks may be read (all 512 blocks) using the switch combination /NFS/ALL (typed in exactly that form with no embedded spaces).

The /DT switch may be used for non file structured transfers.

Directory listing

The command is DDx:/DI or Filename = DDX:/DI. A sample output is as follows

Di	rectory f	or DD1	: 12-Nov-82				
RS	TS/E+ For	mat 4	Directory	segments 1	Open		
RS	TS V7.2-0)4 Mari	ne Labs				
Fi	le Name	Size	Account	Date	Prot	RTS	Pos
ED	T .TSK	A 36	[1,2]	05-0ct-82	<232>	RSX	14
ED	T .SEC	A 122	[1,2]	04-Oct-82	<104>	RSX	50
ED	T .BAC	11	[1,2]	27-Aug-81	<232>	BASIC	172
VT	PLOT.BAS	14	[10,41]	21-0ct-82	<60>	BASIC	183
FN	SC .BAS	1	[1,41]	23-Mar-82	<60>	BASIC	197
<	UNUSED >	10					198
SU	PER .BAS	4	[10,41]	02-Aug-79	<60>	BASIC	208
<	UNUSED >	3					212
SC	RUB .BAS	1	[10,41]	18-Feb-80	<60>	BASIC	215
CH	AMAK.BAS	2	[10,41]	11-Mar-80	<60>	BASIC	216
CH	ART .BAS	1	[10,41]	07-Apr-80	<60>	BASIC	218
SE	EDIR.BAS	1	[1,41]	27-0ct-82	<60>	BASIC	219
SE	EMOR.BAS	1	[1,41]	22-0ct-82	<60>	BASIC	220
SE	EFIL.BAS	1	[1,41]	22-0ct-82	<60>	BASIC	221
SA	TCON.BAS	2	[1,41]	07-0ct-82	<60>	RSX	222
SA	TAN .BAS	2	[1,41]	07-Oct-82	<60>	RSX	224
SA	TYR .BAS	2	[1,41]	08-Nov-80	<60>	BASIC	226
SE	GDIR.BAS	4	[1,41]	18-Apr-80	<60>	BASIC	228
<	UNUSED >	280					232

The contents of the directory listing are explained as follows. The + sign after RSTS/E indicates that space for file attributes has been allocated. The letter 'A' after the file name indicates that this file has attributes. The unused entries are a necessary part of the RT11 file structure. If a file is added to the tape on an RT11 system the columns for account, protection code and RTS will be blank.

Deleting TU58 files

Any files may be deleted from a TU58 tape if the user has write access to the particular files. Wildcard names or indirect command files may be used. The format of the command is ddx:[ppn]filename/DE. The delete operations are done on the directory only. The detach switch (/DT) may be used.

Special tentative file delete

Occasionally a tentative file may be left on the tape. The

directory listing will show these marked with an asterisk after the file name. These may be deleted using the command DDx:[*,*]*.*/DE*.

Conditional delete

The files may be deleted selectively using the switch /IN/DE (in that order). Each eligible file name will be displayed on the terminal and will be deleted if Y is typed and not if N (or anything else) is typed. In this case the detach switch may not be used.

Squeezing the tapes

If files have been deleted there may be room on the tape but not enough contiguous space for a new file, or not enough directory space. If this happens the tape must be squeezed using the command DDx:/SQ/DT. It is advisable to run this detached because it takes time and < cntr>C from the terminal would result in an incomplete data set.

Another method of squeezing is to copy [*,*]*.* in a privileged account to a newly initialized tape. This has the advantage that the data will not be lost in the event of a system crash during the squeeze, but the operation is privileged and uses both units.

Notes on the operation of KIM

Except for non file structured tapes KIM uses an RT11 file structure. Before an operation the tape directories are cached on the system account [1,2] so that it is unnecessary to rewind the tape every time the program requires access to the directory. If the directory is not altered the cache file is [1,2]CAjjDx.TMP where jj is the job number and x is the device number, and this file is deleted on exit from the program. If, however, the directory is altered the cache file takes the form [1,2]????nn.CCH and the tape is marked with this filename. On exit from the program the directory is written back to the tape. It is important, therefore, that the tape is not removed from the drive until the program is complete and the "Ready" message is printed at the terminal or in the case of detached jobs after the finished message is ge is sent to the terminal or the job is logged out. If due to a system failure or to the tape being removed, the directory is not restored the tape will not be readable on another system but as long as the cache file is still in existence the tape can be again used with KIM on the system from which it was illegally dismounted and the directory will be restored on exit.

The program may be terminated with <cntr>C in which case the current operation will be aborted or <cntr>Z which will terminate only when the current operation (including indirect command files) is complete. If the job is detached the program will terminate at the end of the current operation. In order to save directory shuffling and tape movement use wildcard specifications and indirect command files where possible, and continue the next operation without using <cntr>C.

Only one user can access a TU58 device at one time but one user may access DDO: while another is using DD1:. It is advisable to assign the unit or units required before starting operations but don't forget to deassign them when finished unless running detached. If someone else were to gain ac-

cess to the device you are using your program would go wrong and the directory could be destroyed.

Operating instructions

Prepare any indirect command files.

Have the tape mounted (and zeroed if necessary).

Assign the device (DDO: and/or DD1:).

RUN & KIM

Type command.

On prompt (?) type a further command or $\dagger C$ or $\dagger Z$ unless detached.

Deassign the device(s) unless detached.

Tell the operator when you have finished with the tape.

Use wildcards and indirect command files where possible, especially when running detached in order to minimize the physical reading and writing of the directories.

Program listing

```
1 EXTEND
100 DIR.FLAG$=0$\DE.FLAG$=0$\ZE.FLAG$=0$
110 SQ.FLAG$=0$\JSWIT$=0$
120 DIM M$(60), M$(30), M$2(30$), SX$(30), ATR$(16)
130 DIM #3$, RT.HEAD(4$)=512, CACHE$(64$)=512
\DIM #3$, BLANK$(5$,255$), DIR$(30$,511$)
140 S$=SYS(CHR$(6$)+CHR$(-7$)+STRING$(28$,0$))
  150 CMDFIL$=0$
  160 B$=SYS(CHR$(6%)+CHR$(14%)+STRING$(28%.0%))
  CURPPN$-MID(B$,7$,2$)\JOB$-ASCII(B$)?$
170 P1$-ASCII(RIGHT(CURPPN$,2$))\P2$-ASCII(CURPPN$)
 180 CURPT$=P1$\CURP2$=P2$
190 DROP$=CHR$(6$)+CHR$(-21$)\REGAIN$=DROP$+CHR$(0$)\DROP$=DROP$+CHR$(255$)
200 CURPPN$='['+NUM1$(P1$)+','+NUM1$(P2$)+']'
210 ERROR.POINT$=1$
 220 CCL.FLG%=0%
 220 CCL.FLG$=0$
500 IANALYSE THE COMMAND STRING
510 DIR.FLAG$,DE.FLAG$,ZE.FLAG$,SQ.FLAG$=0$
520 ON ERROR GO TO 2050
530 IF CMDFIL$ GO TO 2030
540 IF JSW1T$ THEN OUT.MESS$='KIM Completed at '+TIME$(0)\GO TO 14000
550 IF CCL.FLG$ GO TO 14000
 556 INPUT LINE C$
570 IF CVT$$(C$,63$)='' GO TO 56(
580 COMMAND$=LEFT(C$,LEN(C$)-2$)
\ INP$,OUT$=0$\$WIT$=''
 OUTFILE$, INPFILE$=''
DE.FLAG$=0$
590 J$=LEFT(COMMAND$, 1$)
 | IF J$='\epsilon' GO TO 2000
| F J$<'\' GO TO 620
| F J$<\' GO TO 620
| F$=INSTR(1$,COMMAND$+'=','=')
| SWIT$=SWIT$+LEFT(COMMAND$,P$)
 \ COMMAND$=RIGHT(COMMAND$, P$*1$)
\ IF COMMAND$='' THEN 4000 ELSE GO TO 590
620 IF J$='=' GO TO 3010
630 IF J$=',' THEN ERM$=98$\ERR.L$=LINE\GO TO 9110
  640 IF INSTR(1%, COMMAND$, '/DE') THEN DE.FLAG%=-1%\GO TO 3020
 1000 1
          GET OUTPUT FILE
!OUTPUT PPN CAN BE [#,*] BUT NOT [*,n] nor [n,*]
                  GOSUB 10000
 1020 IF ((M%(29%)AND 128%)=0%) THEN
P1%=CURPIX\P2%=CURPZ%\PPN$=CURPPN$
1030 0UTP1%=P1%\01P2%=P2%
\ 0UTPP.WILD%=(P1%=255%) OR (P2%=255%)
           OUTSPEC% = OUTPP.WILD% AND (P1%<>P2%)
 1040 OUTNAME $=FILE. NAME $+EXTN $
           OUTFIL. WILD$=M$(29$) AND 102$
          OUTSPECS OR(OUTSPECS OR(OUTSPIL WILDS AND (OUTNAME$ <> '??????.???'))

IF CVT$$(OUTNAME$,63$)='' THEN OUTFIL.WILDS=-1$

OUTFILE$=FILE$\OUTSPILE$ OUTSPILE$

OUTFILE$=FILE$\OUTSPILE$

OUTFILE$-FILE$\OUTSPILE$
 1060 IF R$ THEN 590 ELSE GO TO 4000
 2000
          INDIRECT COMMAND FILE ANALYSIS
2010 COMMAND$=RIGHT(COMMAND$,2$)\CMDFIL$=CMDFIL$+1$
        GOSUB 10000
EXTN$='.CMD' IF EXTN$=''
CHAN$=CMDFIL$+7$
        F$=FIL$+PPN$+FILE.NAME$+EXTN$
        S$=SYS(REGAIN$)
```

```
2040 IF CVT$$(c$,63$)='' GO TO 2030
2050 IF CRC*>115 GO TO 9000
2070 IF ERL*>50 THEN RESUME 9150
2060 IF ERL*>50 THEN RESUME 9150
2090 IF CMDFIL$ THEN CLOSE $CHAN$

CMDFIL$=CMDFIL$_-1$_\CHAN$=CHAN$_-1$$
2100 GO TO 500
3000 !

GET INPUT FILE

I

GET INPUT FILE

1

1

3010 COMMAND$=RIGHT(COMMAND$,2$)
3020 GOSUB 10000

INPFILE$=FILE$_\INP$=1$

3030 INPI$=PI$_\INPE$=$2$_\INP, WILD$=(PI$=255$) OR (P2$=255$)

MI$[U$]=30$_\INF[U$]=3$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$]=5$_\INF[U$
```

DEC

SYSTEMS & COMPONENTS

C.D. SMITH & ASSOCIATES, INC. 12605 E. Freeway, Suite 318 Houston, TX 77015 (713) 451-3112

,

CIRCLE 54 ON READER CARD

NEW—comprehensive guide to RSTS software.

THE SIERRA DIRECTORY: RSTS SOFTWARE

The latest, most complete list of RSTS software. Hundreds of software systems and software vendors exclusively for the RSTS Operating System. Product descriptions, prices, technical specifics, vendor support provisions, vendor addresses and phone numbers. Cross-indexed by software function and vendor.

- Financial modeling
- Accounting
- Statistics
- Electronic mail
- Word processing
- Languages and compilers
- Communications
- System utilities
- Programmers tools
- Database systems
- Report writers
- Graphics
- Many industry-specific systems—and much more

ORDER YOUR COPY TODAY FOR PROMPT DELIVERY

Sierra Directory, Dept. 11, 3304 Springhill Rd., Lafayette, CA 94549 (415) 284-1610

Payment is enclosed for	_copies of the Sierra Directory of RSTS				
Software at US\$29.00 each	(US\$31.00	outside	North	America).	
including shipping and handling	ng.				

NAME

ADDRESS

CITY/STATE/ZIP

COUNTRY

RSTS is a trademark of Digital Equipment Corp.
CIRCLE 192 ON READER CARD

	IF OUTSPEC\$ THEN ERM\$=98\$\ERR.L\$=LINE\ GO TO 9110		6010 IF OUTPP.WILD\$ THEN OUTPPN\$=PPN\$	£
	INDEV \$=FIL\$\INPPN\$=PPN\$\INNAME\$=FILE.NAME\$+EXTN\$ IF R\$ GO TO 590		\ OUTP1%=INP1%\OUTP2%=INP2% 6020 IF OUTFIL.WILD% THEN OUTNAME %=IN.FILE\$	
4000	1	& &	6030 EMPTY.SEG\$,TMP.SEG\$=-1\$ 6040 IF LEFT(OUTDEV\$,2\$)='DD' GO TO 6500	
1	DADGUMD WID COMMAND LYND	& &	6050 1	ė.
1	EXECUTE THE COMMAND LINE		! RSTS OUTPUT FILE !	œ
4005	IF (LEFT(OUTDEV\$,2\$)<>'DD') AND (LEFT(INDEV\$,2\$)<>'DD') THEN OUT.MESS\$='Either input or output must be DD:'\GOSUB 18000\GO TO 500	&	6060 IF OUTDEV\$='KB:' GO TO 6210 6070 OUTREC\$=1\$	
	IF SWIT\$='' GO TO 4070 IF LEFT(SWIT\$,3\$)='/ZE' GO TO 4070		6080 X\$=OUTDEV\$+OUTPPN\$+OUTNAME\$	
4030	IF JSWIT# GO TO 4050		6090 IF ATR\$(1\$)=29580\$ THEN PROT\$='<'+NUM1\$(ATR\$(3\$))+'>' ELSE PROT\$='' 6100 IF INSTR(1\$,SWIT\$,'/NS')=0\$ GO TO 6120	
4040	JSWIT\$=INSTR(1\$,SWIT\$,'/DT')\ IF JSWIT\$ THEN SWIT\$=LEFT(SWIT\$,JSWIT\$-1\$)+RIGHT(SWIT\$,JSWIT\$+3\$)	Ł	6110 S\$=SYS(DROP\$)\OPEN X\$ FOR INPUT AS FILE #1\$\CLOSE #1\$ \ S\$=SYS(REGAIN\$)\GO TO 6610	&
4050	IF INSTR(1%, 'DE, DE*, DI, SQ, NS, NFS, NFS/ALL, IN/DE', RIGHT(SWIT\$, 2\$)) GO TO 4070	ě	6120 S\$=SYS(DROP\$) 6125 IF NFS\$ THEN FILZ\$=4\$ ELSE FILZ\$=INSIZE\$	
	OUT.MESS\$='Illegal switch'\GOSUB 18010\GO TO 500		6130 OPEN X\$+PROT\$ FOR OUTPUT AS FILE #1%, RECORDSIZE 512%, FILESIZE FILZ%	&
	NFS\$=INSTR(1\$,SWIT\$,'/NFS') ALL\$=INSTR(1\$,SWIT\$,'/ALL')		\	&
	IF NFS% AND (LEFT(OUTDEV\$,2%)='DD') GO TO 7190 IF NFS% GO TO 4190		\ IF DATE.RSTS\$=-1\$ THEN J\$=CHR\$(8\$) ELSE J\$=CHR\$(12\$) 6150 S\$=SYS(CHR\$(6\$)+CHR\$(-26\$)+CHR\$(1\$)+J\$+STRING\$(8\$,0\$)	&
4100	POINTERS=INSTR(1\$,SWIT\$,'/ZE')		+CHR\$(ATR\$(4\$))+CHR\$(SWAP\$(ATR\$(4\$)))	<u>.</u>
4120	IF POINTER\$ GO TO 12010 POINTER\$=INSTR(1\$,5*IT\$,'/DI')\IF POINTER\$ THEN GOSUB 24000 !DIRECTORY		+CHR\$(ATR\$(5\$))+CHR\$(SWAP\$(ATR\$(5\$)))+STRING\$(2\$,0\$) +CHR\$(DATE.RSTS\$)+CHR\$(SWAP\$(DATE.RSTS\$))	& &
4130 4140	IF INSTR(1%, SWIT\$, '/SQ') THEN SQ.FLAG\$=2%\ GO TO 14000	&	+CHR\$(40\$)+CHR\$(111\$)+STRING\$(8\$,0\$)) 6160 IRESTORE ATTRIBUTES	
1	GET DIRECTORIES AND CACHE IF DD:	Ł	6170 FOR J\$=16\$ TO 6\$ STEP -1\$\IF ATR\$(J\$)<>0\$ GO TO 6190	Ł
	INDIR\$=''\OUTDIR\$=''	&	GO TO 6200 INO ATTRIBUTES	
/	IF LEFT(OUTDEV\$,2\$)='DD' THEN DDDEV\$=OUTDEV\$\INPUT.FLAG\$=0\$ GOSUB 8000\ OUTDIR\$=CACHE.TMP\$(IDEV\$)	Ł	6190 N.ATT%=J%-5% \ S\$=CHR\$(6%)+CHR\$(-25%)+CHR\$(1%)+CHR\$(N.ATT%)	& &
4160	IF LEFT(INDEV\$,2%)='DD' THEN DDDEV\$=INDEV\$\INPUT.FLAG%=-1% GOSUB 8000 \INDIR\$=CACHE.TMP\$(IDEV%)	&	\ S\$=S\$+CHR\$(ATR\$(J\$))+CHR\$(SWAP\$(ATR\$(J\$))) FOR J\$=6\$ TO 16\$	&
4170	IF DIR.FLAGS GO TO 24040		\ s\$=SYS(S\$+STRING\$(4\$,0\$)) 6200 GO TO 7000	
	IF SQ.FLAG\$ GO TO 17000 SUCCESS\$=0\$\INDEXP\$=0\$\INPUT.INDEX\$=0\$		6210 OPEN OUTDEY\$ AS FILE #1%, RECORDSIZE 512% \ IF NFS% GO TO 6230	Ł
4200	INDEX%=INPUT.INDEX% IF LEFT(INDEV\$,2%)='DD' GO TO 5500	&	6220 PRINT\PRINT 'File '; PPN\$; IN. FILE\$\PRINT 6230 OUTREC\$=4096\$\ GO TO 7000	
4210	IF SQ.FLAG\$ GO TO 5500	Ł	6500 I	ě.
5000 !	1	å	! RT11 GET SPACE FOR FILE	å
1	FIND A MATCHING ACCOUNT AND FILE IF RSTS	å å	6510 IF NFS\$ AND (LEFT(OUTDEV\$,2\$)='DD')THEN OUTREC\$=1\$\ GO TO 14010 6520 OPEN OUTDIR\$ FOR INPUT AS FILE #3\$	
1			6530 TYPE\$=4\$\F\$=IN.FILE\$\INDEX\$=0\$\GOSUB 27000	&
5010	SX\$(I\$)=M1\$(I\$) FOR I\$=0\$ TO 30\$ SX\$(2\$)=25\$	& &	\ IF SEG\$<0\$ GO TO 6620 6540 IF DIR\$(SEG\$,ENT\$+6\$)<>29580\$ GO TO 6600	
\	SX\$(3\$)=INDEXP\$ AND 255\$\SX\$(4\$)=SWAP\$(INDEXP\$) AND 255\$ SX\$(5\$)=INP2\$\SX\$(6\$)=INP1\$	& &	6550 DIR.P1%=SWAP%(DIR%(SEG%,ENT%+7%)) AND 255% \DIR.P2%=DIR%(SEG%,ENT%+7%) AND 255%	&
\	SX\$(I\$)=0\$ FOR I\$=7\$ TO 22\$	&	\ IF(OUTP1%<>DIR.P1%) OR (OUTP2%<>DIR.P2%) GO TO 6620	&
\ 5020	CHANGE SX\$ TO SX\$ SX\$=SYS(SX\$)		6560 ERM\$=10\$\ERR.L\$=LINE \ PROT\$=DIR\$(SEG\$,ENT\$+8\$) AND 63\$\IF CURP1\$=1\$ GO TO 6600	u
	CHANGE SX\$ TO SX\$ ACT.P1\$=SX\$(6\$)\ACT.P2\$=SX\$(5\$)	&	6570 IF PROT\$ AND 1\$ GO TO 9110 6580 IF (PROT\$ AND 20\$) AND (OUTP1\$<>CURP1\$) GO TO 9110	
\	M1%(3%)=INDEX% AND 255%\M1%(4%)=SWAP%(INDEX%) AND 255%	& &	6590 IF (PROT% AND 4%) AND (OUTP2%<>CURP2%) GO TO 9110	
1	M1%(I%)=SX%(I%) FOR I%=5 TO 6 CHANGE M1% TO M\$	•	6600 TMP.SEG\$=SEG\$\TMP.ENT\$=ENT\$ 6610 IF INSTR(1\$,SWIT\$,'/NS') THEN	٠.
5050 5060	M\$=SYS(M\$) CHANGE M\$ TO M2\$	å	PRINT IN.FILE\$+' Exists'UNLESS JSWIT\$ \ GO TO 7180	ě.
1	P1%=M2%(6%)\P2%=M2%(5%)\PROT%=M2%(15%)	ě L	6620 IGET EMPTY SEGMENT	&
'	M2\$(I\$)=M2\$(I\$)+SWAP\$(M2\$(I\$+1\$)) FOR I\$=5\$ TO 19\$ STEP 2\$ DATE.RSTS\$=M2\$(19\$)\GOSUB 15000 IGET RT11 DATE	Ł	INDEX\$=0\$ 6630 TYPE\$=2\$\SIZE\$=INSIZE\$	<u>.</u>
5070 \	IN.FILE\$=RAD\$(M2\$(7\$))+RAD\$(M2\$(9\$))+'.'+RAD\$(M2\$(11\$)) PPN\$='['+NUM1\$(P1\$)+','+NUM1\$(P2\$)+']'	a &	\ GOSUB 27000 \ IF SEG\$<0\$ GO TO 6650	۵
\ E080	INSIZE(=M2\$(13\$)\SUCCESS\$=1\$ XIN\$=INDEV\$+PPN\$+IN.FILE\$	Ł	6640 EMPTY.PNT%=PNT%\EMPTY.SEG%=SEG%\EMPTY.ENT%=ENT% 6650 IIS THERE ROOM IN THIS SEGMENT	&
\	S\$=SYS(DROP\$)	å	TYPE\$=2\$\SIZE\$=-1\$\INDEX\$=-(SEG\$+1\$)\GOSUB 27000	
'	OPEN XIN\$ FOR INPUT AS FILE #2% S\$=SYS(REGAIN\$)	å	6660 ENTRY.LEN%=7%+DIR%(0%,3%)/2% 6670 IF (ENT%+ENTRY.LEN%)>511% THEN EX.SEG%=SEG%	&
\	ATR\$=SYS(CHR\$(6\$)+CHR\$(-25\$)+CHR\$(2\$)+CHR\$(0\$)+STRING\$(26\$,0\$)) CHANGE ATR\$ TO M2\$	& &	\ GOSUB 25000\ GO TO 6620 6680 IF EMPTY.SEG\$<0\$ THEN ERM\$=4\$\ERR.L\$=LINE\GO TO 9110	
· ·	ATR\$(O\$)=DATE.RT11\$	& &	6690 DIR\$(EMPTY.SEG\$,EMPTY.ENT\$+3\$)=DIR\$(EMPTY.SEG\$,EMPTY.ENT\$+3\$)-INSIZE\$	å
'	ATR\$(1\$)=29580\$ ATR\$(2\$)=P1\$#256\$+P2\$	&	FOR IS=LIS TO EMPTY. ENTS STEP -1%	&
\	ATR\$(3\$)=PROT\$ ATR\$(1\$)=M2\$(2\$*1\$+19\$)+SWAP\$(M2\$(2\$*1\$+20\$))FOR I\$=4\$ TO 5\$	& &	NEXT IS	å
1	ATR\$(I\$)=M2\$(2\$*I\$-7\$)+SWAP\$(M2\$(2\$*I\$-6\$))FOR I\$=6\$ TO 16\$ CLOSE #2\$	&	6700 DIR\$(EMPTY.SEG\$,EMPTY.ENT\$-1\$)=256\$ M\$=SYS(CHR\$(6\$)+CHR\$(-23\$)+OUTNAME\$)	&
5090	INRECS=15		6710 CHANGE M\$ TO M\$	& &
5100	IF DE.FLAG\$ THEN OUT.MESS\$="This program can't delete a disk file" GOSUB 18010\GO TO 500	&	DIR*(EMPTY.SEG\$,EMPTY.ENT\$+1\$)=M\$(1\$*2\$+7\$) FOR 1\$=0\$ TO 2\$	&
5110 5500	GO TO 6000	Ł	\ DIR4(EMPTY.SEG%.EMPTY.ENT%+3%)=INSIZE%	& &
1		& &	\ I.IMIT%=DIR%(0%,3%)/2%	ě
1	GET INPUT FILE FOR RT11 DEVICE	ě.	IF BLANK\$(1\$,235\$)<>29580\$ THEN LIMIT\$=0\$ 6720 DIR\$(EMPTY.SEG\$,EMPTY.ENT\$+5\$+1\$)=ATR\$(1\$) FOR 1\$=0\$ TO LIMIT\$	
! 5510	XIN\$=INDEV\$		6730 IF OUTRECS+INSIZE\$>511\$ THEN ERM\$=4\$\ERR.L\$=LINE\GO TO 9110 6740 OPEN OUTDEV\$ AS FILE \$1\$	
5520	IF NFS% THEN INSIZE%=-1%\INREC%=1%\GO TO 6000 IF INSTR(1%,SWIT\$,'/DE*') THEN TP%=1% ELSE TP%=4%		7000 1	& &
5540	OPEN INDIR\$ FOR INPUT AS FILE #3%	&	! COPY THE FILES	å
	TYPE%=TP%\F%=INNAME%\GOSUB 27000 IF SEG%<0% GO TO 7230 !CANT FIND FILE OR ACCOUNT	&		
5550	IGET THE EXTRA WORDS IF ANY ATR\$(1\$)=0\$\LIMIT\$=DIR\$(0\$,3\$)/2\$	&	7010 CNT%=0%\S\$=SYS(DROP\$)UNLESS SQ.FLAG%	ě.
\	FOR I%=0% TO 16%	&	OPEN XIN\$ FOR INPUT AS FILE #2% S\$=SYS(REGAIN\$)UNLESS SQ.FLAG\$	Ł
1	ATR\$(I\$)=DIR\$(SEG\$,ENT\$+5\$+I\$) UNLESS I\$>LIMIT\$ NEXT I\$	&	7020 IF JSWIT\$ >0\$ THEN PRINT 'Detaching' JSWIT\$=-1\$\S\$=SYS(CHR\$(6\$)+CHR\$(7\$)+CHR\$(128\$))	Œ
	IF SQ.FLAG\$ GO TO 5670 IF ATR\$(1\$)<>29580 GO TO 5670		7025 IF INSIZE#=0# GO TO 7100	&
5590	DIR.P1%=(SWAP%(ATR%(2%)) AND 255)\DIR.P2%=(ATR%(2%) AND 255)		7030 FIELD #2%,512% AS Q2% \ HUNG%=0%	
5610	IF (INP2\$<>255) AND (INP2\$<>DIR.P2\$) GO TO 7180 IF (INP1\$<>255) AND (INP1\$<>DIR.P1\$) GO TO 7180		7040 GET #2\$, RECORD INREC\$, COUNT 512\$ 7042 IF NFS\$=0\$ GO TO 7050	
5620 5630	IF DE.FLAG\$ THEN Y\$=1\$ ELSE Y\$=2\$ ERM\$=10\$\PROT\$=DIR\$(SEG\$,ENT\$+8\$) AND 63\$\IF CURP1\$=1\$ GO TO 5670		7045 IF LEFT(INDEV\$,2\$)='DD' AND Q2\$=STRING\$(512\$,0\$) GO TO 7120 UNLESS ALL\$ 7050 FIELD \$1\$,512\$ AS Q1\$	Ł
5640	IF PROT\$ AND (15*Y\$) GO TO 9180 IF (PROT\$ AND (205*Y\$)) AND (DIR.P1\$<>CURP1\$) GO TO 9180		\ LSET Q1\$=Q2\$	å
5660	IF (PROTS AND (45#YS)) AND (DIR.P2\$<>CURP2\$) GO TO 9180		\ HUNG\$=4\$ 7060 PUT #1\$, RECORD OUTREC\$, COUNT 512\$	&
5680	INSIZE = DIR (SEG *, ENT *+3 *) \ IN. FILE *= ACT. FIL * \ INREC *= PNT * \ SUCCESS *= 1 * IF SQ. FLAG *=-1 * THEN INREC *= INREC *+1 *		CNT%=CNT%+1%\INREC%=INREC%+1% OUTREC%=OUTREC%+1% UNLESS OUTDEV%='KB:'	& &
5690	IF DE.FLAG\$ THEN GOSUB 13000\GO TO 4200 CLOSE #3\$		\ IF (CNT%=INSIZE%) GO TO 7100	
6000		& &	7090 GO TO 7030 7100 IF NFS\$=0\$ GO TO 7120	ě.
! !	GET OUTPUT FILE	&	7110 IF (LEFT(OUTDEY\$,2\$)='DD') AND (NFS\$>0\$) THEN Q2\$=STRING\$(512\$,0\$) \[\text{INSIZE}=INSIZE\$+1\$\NFS\$=-1\$\G0 TO 7050	u.
! !		&	7120 CLOSE #1\$, #2\$ 7130 IF NFS\$ GO TO 500	
			1130 II 1120 GO 10 300	

RSTS/E INTERNALS MANUAL

Second Edition — Including Version 8.0

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 8.0 and earlier versions of the monitor. Future updates will include changes for new versions of the monitor.

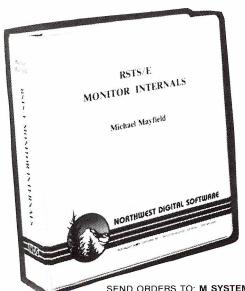
Information is available for all levels of users:

- 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 five quick reference foldout charts:

- 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

```
7140 IF SQ.FLAG$=1$ GO TO 17100
7150 OPEN 'KB:' AS FILE #4$, MODE 16$ UNLESS JSWIT$<0$
7155 IF EMPTY.SEG$>=0 THEN DIR$(EMPTY.SEG$, EMPTY.ENT$-1$)=1024$
                                                                                                                                                                                                                                                                                                                                                                                                   10090 IF (M$(28$) AND 2$) = 0$ THEN EXTN$=''
ELSE EXTN$='.'-RAD$(M$(11$))
1010 FILE.NAME$=RAD$(M$(7$))+RAD$(M$(9$))
10110 IF M$(28$) AND 16$ THEN FIL$=DEVICE$+UNIT$+':' ELSE FIL$=''
10120 FILE$=FIL$+PPN$+FILE.NAME$+EXTN$\RETURN
     7160 IF TMP.SEG$>=0 THEN SEG$=TMP.SEG$\ENT$=TMP.ENT$\GOSUB 13000
   7170 CLOSE #4$
7180 IMPUT.INDEX$=IMPUT.INDEX$+1$\GO TO 4200
   SWITCH ANALYSIS
                                                                                                                                                                                                                                                                                                                                                                                                   1010 THIS.SWITCH$=RIGHT(SWIT$, POINTER$)
11020 P$=INSTR(2$, THIS.SWITCH$+'','|')
11030 O$=INSTR(2$, THIS.SWITCH$+'=','=')
11040 IF Q$<P$ THEN P$=Q$
11050 SWIT$=LEFT(SWIT$, POINTER$-1$)+RIGHT(THIS.SWITCH$, P$)
11060 THIS.SWITCH$=LEFT(THIS.SWITCH$, P$-1$)
11070 RETURN
12000 |
                       CACHE THE DIRECTORY -ONLY FOR DD:
| 8010 IF DE.FLAG$ THEN INPUT.FLAG$=0$
8020 ICHECK IF DIRECTORY IS ALREADY CACHED
8030 Z$=MID(DDDEV$,3$,1$)\Z$='0' IF Z$=':'
OPEN DDDEV$,3$,1$)\Z$='0' IF Z$=':'
OPEN DDDEV$,3$,1$)\Z$='0' IF Z$=':'
OPEN DDDEV$,3$,1$)\Z$='0' IF Z$=':'
OPEN DDDEV$,3$,1$)\Z$='0' IF Z$=':'
OPEN DDEV$,4L(Z$)\IF CACHE$(IDEV$) GO TO 8110
8040 IGET THE FIRST DIRECTORY BLOCK IF FIRST WORD IS NEGATIVE
ITRY TO FIND THE CACHE FILE, ERROR IF NOT FOUND, IF FOUND
IOPEN IT INSTEAD OF THE TEMP FILE
8050 REC$=6$\GSUB 20000
DIR.LEM$=SWAP$(CYT$$(LETT(BUPIN$,2$)))
FIRST.DIR$=BUPIN$+''
HISEG.OPEN$=SWAP$(CYT$$(MID(FIRST.DIR$,5$,2$)))
IF DIR.LEM$<OS THEN NAM$=255$-(SWAP$(DIR.LEM$) AND 255$)
NAM$=''\NAM$=MIDHENT$,3$,7$,9$ FOR J$=4$ TO 10$ STEP 2$
CACHE.THP$(IDEV$)='1,2''+NAM$+NUM1$(NAM$)+'.CCH'
OPEN CACHE.TMP$(IDEV$) FOR INPUT AS FILE #3$
                                                                                                                                                                                                                                                                                                                                                                                                      12000 1
                                                                                                                                                                                                                                                                                                                                                                                                                           ZERO A DD: DEVICE
                                                                                                                                                                                                                                                                                                                                                                                                      12010 IF CURP1$<>1$ THEN ERM$=10$\ERR.L$=LINE\GO TO 9110
                                                                                                                                                                                                                                                                                                                                                                                                    12020 IF DEVICE$<>'DD' THEN OUT.MESS$='Invalid device'\GOSUB 18010
                                                                                                                                                                                                                                                                                                                                                                                                    GO TO 500

12030 IF PPN$<>CURPPN$ THEN OUT.MESS$='No account number allowed'
                                                                                                                                                                                                                                                                                                                                                                                               GO TO 500
12030 IF PPN$</UNIPPN$ THEN OUT.MESS$='No account number allowed'
GOSUB 18010GO TO 500
12040 IF FILE.NAME$+EXTN$<'' THEN OUT.MESS$='No filename allowed'
GOSUB 18010GO TO 500
12050 GOSUB 11010 IEXTRACT THE SWITCH
12060 REST$=RIGHT(RHIS.SWITCH$,#$)
12070 IF LEFT(REST$,1$)='+' THEN EXTRA.WORDS$=16$\GO TO 12100
12080 EXTRA.WORDS$=5\GO TO 12110
12100 REST$=RIGHT(REST$,2$)
1210 IF LEFT(REST$,1$)='-' THEN DIR.LEN$=VAL(RIGHT(REST$,2$))\GO TO 12140
12100 FEXTRA.WORDS$=5\GO TO 12110
12100 REST$=RIGHT(REST$,2$)
12110 IF LEFT(REST$,1$)='-' THEN DIR.LEN$=VAL(RIGHT(REST$,2$))\GO TO 12140
12120 IF EXTRA.WORDS$=16$ THEN DIR.LEN$=0$ ELSE DIR.LEV$=4$
12130 IF REST$<'' THEN OUT.MESS$='Directory too long'
GOSUB 18010\GO TO 500
12150 FIRST.FILES$=2$*DIR.LEV$+6$
12160 ROOM$=512$-FIRST.FILES$
12160 ROOM$=512$-FIRST.FILES$-1
12170 PRINT'Really initialise ';DEVICE$;UNIT$;':'
12180 INPUT Y$
12190 IF LEFT(CVT$$(Y$,63$),1$\()<'Y'$ GO TO 500
12200 OPEN DEVICE$+UNIT$*-'' AS FILE $1$
12210 IF INSTR(1$,SWIT$,'\RT11')=0$
THEN RST$=CVT$$(-29561$) ELSE RST$=STRING$(2$,0$)
12200 THEN RST$=CVT$$(-29561$) ELSE RST$=STRING$(2$,0$)
12200 GRANGE HOME$ TO M$
SUMCH$=0$
SUMCH$=5UGH$+H$(1$)+CHR$(0$)+CHR$(0$)+RST$
+HID (SYS(CHR$(6$)+CHR$(9$)+CHR$(0$)),3$,24$)+'DECRT11A

SUMCH$=0$
SUMCH$=5UGH$+H$(1$) FOR I$=1$ TO M$(0$)
                       CACHES(IDEVS)=15
  CACHES(IDEV$)=1$
GO TO 8110

8060 CACHE.THP$(IDEV$)='[1,2]CA'+NUM1$(JOB$)+'D'+NUM1$(IDEV$)+'.TMP'
OPEN CACHE.TMP$(IDEV$) FOR OUTFUT AS FILE #3$
8070 FOR J$=1$ TO HISEG.OPEN$
FOR I$=0$ TO 1$
REC$=J$=2*+#$+1$ GOSUB 20000
CACHER (ASCA) FOR ASCA PERNAL!
                       CACHE$(J$#2$+I$-1$)=BUFIN$+1
                       INDEXS=-J$\TYPE$=2$\SIZE$=-1$\GOSUB 27000
  8140 RANDOMIZE
                       NAM$=''\NAM$=NAM$+CHR$(INT(RND(0)#25$+65$)) FOR K$=1$ TO 4$
                                                                                                                                                                                                                                                                                                                                                                                                                           SUMCHS = 05
                      NAMS=RND(0)*99

CACHE.FIL$=[1,2]'+NAM$+NUM1$(NAM$)+'.CCH'

OPEN CACHE.FIL$ FOR INPUT AS FILE #7$\CLOSE #7$
                                                                                                                                                                                                                                                                                                                                                                                                   \SUMCH$=0$\text{SUMCH$$\frac{1}{2}\text{SUMCH$$\frac{1}{2}\text{MCH$$\frac{1}{2}\text{MCH$$\frac{1}{2}\text{MCH$$\frac{1}{2}\text{MCH$$\frac{1}{2}\text{MCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$$\frac{1}{2}\text{MAP}(-SUMCH$\frac{1}{2}\text{MAP}(-SUMCH$\frac{1}{2}\text{MAP}(-SUMCH$\frac{1}{2}\text{MAP}(-SUMCH$\frac{1}{2}\text{MAP}(-SUMCH$\frac{1}{2}\text{MAP}(-SUMCH$\frac{1}{2}\text{MAP}(-SUMCH$\frac{
  \ GO TO 8130
8150 PART1$=LEFT(FIRST.DIR$,10$)\PART2$=RIGHT(FIRST.DIR$,11$)
                       CHANGE PART1$ TO M%
                       M$(2$) =- NAM$
                     M$(2$)=-NAM$

M$(2$)=-NAM$

M$(2$(1$+K$))=ASCII(MID(NAM$,K$,1$)) FOR K$=1$ TO 4$

CHANGE M$ TO PARTI$

FIELD $2$,512$ AS A$\HUNG$=4$

LSET A$=PARTI$+PART2$

PUT $2$, RECORD 6$, COUNT 512$

NAME CACHE. THP$(IDEY$) AS CACHE.FIL$

CACHE. TMP$(IDEY$)=CACHE.FIL$
                                                                                                                                                                                                                                                                                                                                                                                                                           M$(I$)=0$ FOR I$=1$ TO JEX$
                                                                                                                                                                                                                                                                                                                                                                                                                           H$(0$)=JEX$

H$(0$)=JEX$

H$(1$)=DIR.LEN$

H$(5$)=1$

H$(7$)=EXTRA.WORDS$*2$
                                                                                                                                                                                                                                                                                                                                                                                                                            M$(9$)=FIRST.FILES$
M$(12$)=2$
M$(13$)=213$
CACHE. TMP%(IDEW$)=CACHE.FIL$
8170 CACHES(IDEW$)=1$
8180 CLOSE $2$, $3$\RETURN
9000 IERROR ROUTINES
1 END OF RSTS/E DIRECTORIES
9010 IF (ERL=510) THEN RESUME 6130
9020 IF (ERL=6110) THEN RESUME 6130
9030 IF (ERL=5020) AND SUCCESS* THEN RESUME 500
9040 IF (ERL=5020) AND SUCCESS* THEN RESUME 9110
9050 IF (ERL=5050) THEN EMB*=1$\RER.L$=ERL\RESUME 9110
9050 IF (ERL=5050) THEN RESUME 8150
9060 IF (ERL=5050) THEN INDEXP$=INDEXP$+1$
\( INDEX$=0$\X \RESUME 5000
9070 IOTHER ERRORS
9075 IF (ERR=10$) AND (ERL=5080) THEN RESUME 7180
9075 IF (ERR=10$) AND (ERL=5080) THEN RESUME 7180
9090 IF ERR=14 GO TO 9160 IDEVICE HUNG
9095 IF (ERR=27) AND (ERL=7150) THEN RESUME 7155
9100 ERR.L$=ERL\REM$=ERR\RESUME 9110
9110 OUT.MESS$=RIGHT(SYS(CHR$(6$\X')+CHR$(ERM$)),3$\X' At line' +NUH$(ERR.L$)\X GOSUB 18010
    8170 CACHE$(IDEV$)=1$
                                                                                                                                                                                                                                                                                                                                                                                                                            M$(15$)=17$
M$(16$)=103$
M$(17$)=124$
M$(18$)=38$
                                                                                                                                                                                                                                                                                                                                                                                                                           M%(19%)=ROOM% AND 255%
M%(20%)=SWAP%(ROOM%)AND 255%
M%(JEX%)=8%
                                                                                                                                                                                                                                                                                                                                                                                                                                     CHANGE MS TO DIRECTS
                                                                                                                                                                                                                                                                                                                                                                                                    12290
                                                                                                                                                                                                                                                                                                                                                                                                   12290 CHANGE HS TO DIRECTS

DIRECTS-DIRECTS+STRING$(406$,0$)
12300 FIELD #1$,512$ AS A$

LEST A$=DIRECT$

PUT #1$,RECORD 6$,COUNT 512$

LEST A$=STRING$(512$,0$)

PUT #1$, RECORD 7$,COUNT 512$

12310 PRINT DEVICE$+UNIT$+': Initialised'
                                                                                                                                                                                                                                                                                                                                                                                                    12320 CLOSE #1$,2$
12330 ZE.FLAG$=0$\GO TO 14000
                       +NUM$(ERR.L$)\ GOSUB 18010
  9120 CLOSE #15,#25
9130 IF ERMS=105 THEN S$=SYS(REGAIN$)
9140 IF ERROR.POINT$ GO TO 500
                                                                                                                                                                                                                                                                                                                                                                                                                           DELETE A FILE
                                                                                                                                                                                                                                                                                                                                                                                                   9150 GO TO 14000
9160 HUNG$=HUNG$+1$\IF HUNG$<7$ THEN RESUME
9165 SLEEP 4\IF HUNG$<16$ THEN RESUME
  9170 GO TO 9100
9180 IF INPP.WILD$ GO TO 7180
9190 ERR.L$=LINE\GO TO 9110
     10000 1
              FILE NAME STRING SCAN
                                                                                                                                                                                                                                                                                                                                                                                                  M$=OUTPUT STRING
              M$(5$)=PPN
M$(7$),M$(9$)=FILENAME
  | M$(7$),M$(9$)=FILENAME
| M$(11$)=EXTENSION
| M$(2$) TO M$(26$)=DEVICE
| M$(27$) TO M$(26$)=DEVICE
| M$(27$) TO M$(30$)=FLAGS
| 10010 M$=SYS(CHR$(6$)+CHR$(-23$)+COMMAND$)
| 10020 R$=RECOUNTUN$=LEN(COMMAND$)-R$
| SX$(1$)=M$(1$) FOR I$=1$ TO 30$
| P1$=M$(6$)\P2$=M$(5$)
| M$(1$)=M$(1$)+SMAP$(H$(1$+1$)) FOR I$=5$ TO 15$ STEP 2$
| 10040 COMMAND$=RIGHT(COMMAND$,N$=1$)
| UNIT$=''
                                                                                                                                                                                                                                                                                                                                                                                                  DEL.CNT%-DEL.CNT%-JIG$

13080 NEXT.ENTS-ENT%-JIG$

NEXT.SIZE%-DIR%(SEG$,NEXT.ENT%-1%)

NEXT.SIZE%-DIR%(SEG$,NEXT.ENT%-1%)

NEXT.TYPE%-DIR%(SEG$,NEXT.ENT%-1%)

DEL.CNT%-DEL.CNT%-JIG$

NEXT.ENT%-NEXT.ENT%-JIG$

13100 DIR%(SEG$,DEL.ENT%-1%)-512%

DIR%(SEG$,DEL.ENT%-1%)-512%

DIR%(SEG$,DEL.ENT%-1%)-512%

DIR%(SEG$,DEL.ENT%-1%)-26385%

DIR%(SEG$,DEL.ENT%-1%)-25285%

DIR%(SEG$,DEL.ENT%-1%)-25285%

DIR%(SEG$,DEL.ENT%-1%)-25285%

DIR%(SEG$,DEL.ENT%-1%)-25285%

DIR%(SEG$,DEL.ENT%-1%)-25285%

DIR%(SEG$,DEL.ENT%-1%)-25285%

DIR%(SEG$,DEL.ENT%-1%)-25285%
                      UNIT $= ''

IF M$(26$) THEN UNIT $= NUM1 $ (M$(25$))
  \ IF M$(26$) THEN UNIT$=NUMI$(M$(25$))
10050 DBYUG$$=MID(M$,23$,2$)
\ IF M$(5$)=0$ THEN PPN$=CURPPN$\GO TO 10090
10060 IF P1$=255$ THEN P1$="" ELSE P1$=NUM1$(P1$)
10070 IF P2$=255$ THEN P2$="" ELSE P2$=NUM1$(P2$)
10080 PPN$="["+P1$+","+P2$+"]"
                                                                                                                                                                                                                                                                                                                                                                                                                            DIR%(SEG%, DEL.ENT%+3%)=DE.SIZE%
DIR%(SEG%, DEL.ENT%+5%)=O%
```

```
\ DIR$(SEG$,DEL.ENT$+6$)=0$ UNLESS DIR$(0$,3$)=0$
13110 IF DEL.CNT$ =0$ GO TO 13130
13120 FOR I$=NEXT.ENT$-1$ TO 511$
                                  DIR$(SEG$, I$-DEL.CNT$)=DIR$(SEG$, I$)
NEXT I$
  \ NEXT I$

3130 SEGS=0$

13150 TYPE$=2$\SIZE$=-1$\INDEX$=-(SEG$+1$)\GOSUB 27000

13160 IF DIR$(SEG$,ENT$-10$-15)<>512$ GO TO 13280

13170 NEXT.SEG$-DIR$(SEG$,1)-15

13180 IF NEXT.SEG$</br>
13190 IF DIR$(NEXT.SEG$,55)<>512$ GO TO 13270

13250 ADD.LEN$=DIR$(NEXT.SEG$,5)<>512$ GO TO 13270

13250 ADD.LEN$=DIR$(NEXT.SEG$,9$)

DIR$(SEG$,ENT$+3$-JIG$)=DIR$(SEG$,ENT$+3$-JIG$)+ADD.LEN$

DIR$(NEXT.SEG$,X$)=DIR$(NEXT.SEG$,1$)+ADD.LEN$

13260 DIR$(NEXT.SEG$,1$)=DIR$(NEXT.SEG$,1$-JIG$) FOR I$=5$ TO 511$-JIG$

13270 IF DIR$(NEXT.SEG$.65)=2048 THEN DIR$(SEG$.1$-DIR$(NEXT.SEG$.1$)
  13260 DIRK(NEXT.SEG$,I$)=DIRK(NEXT.SEG$,I$+JIG$) FOR I$=5$ TO 511$-JIG$
13270 IP DIRK(NEXT.SEG$,5$)=2048 THEN DIRK(SEG$,1$)=DIRK(NEXT.SEG$,1$)
13280 IP SEG$>-0$ GO TO 13150
13290 IP SEG$>-0$ GO TO 13150
13300 MK(I$)=0$ FOR I$=0$ TO 31$ \MK(I$)=-1$

SEG$=0$
13302 NEXT.SEG$=DIRK(SEG$,I$)\IF NEXT.SEG$=0$ GO TO 13306
13304 MK(NEXT.SEG$)=-1$\SEG$=NEXT.SEG$-1$\G O TO 13302

13306 FOR I$=1$ TO DIRK(O$,O$)

DIRK(O$,O$)=1$ IF MK(I$)

3310 NEXT I$
     13310 NEXT IS
13320 CLOSE #35 RETURN
14000 !
                                    RESTORE THE DIRECTORY AND EXIT
14140 RESUME 14150
14150 ON ERROR GO TO 14340
14160 CLOSE #15, #25
14170 X$='DD1+NUM1$(1$)+':'
14180 OPEN X$ AS FILE #2$
14170 X$='DD1+NUM1$(1$) #25
14190 RECX=6$\(GSUB 20000\)
14200 DIR.LEM$=SMAP$(CVT$$(LEFT(BUFIN$,2$)))
FIRST.DIR$=BUFIN$*-''

IF DIR.LEM$<=SMAP$(CVT$$(LEFT(BUFIN$,2$)))
FIRST.DIR$=BUFIN$*-''

IF DIR.LEM$<=NAM$=NAM$=P256$-(SWAP$(DIR.LEM$) AND 255$)
NAM$=''\NAM$=NAM$+ND(BUFIN$,J$,1$) FOR J$=4$ TO 10$ STEP 2$
CLOSE #3$
OPEN '[1,2]'+NAM$+NUM1$(NAM$)+'.CCH' FOR INPUT AS FILE #3$
FIRST.DIR$=CACHE$(1$)
HISEG.OPEM$=SWAP$(CVT$$(MID(CACHE$(1$),5$,2$)))
FOR K$=1$ TO HISEG.OPEM$=2$
14210 FIELD #2$,512$ AS A$\(NUN0$=4$\)
14220 B$=LEPT(CACHE$(K$)+STRING$(512$,0$),512$)
LSET A$=B$
14210 FIELD $25,512$ AS A$\times \text{A$\times \text{A}\times \te
    \ FIELD #1% 512% AS Q1$\HUNO$=4%\LSET Q1$=STRING$(512%,0%)
14400 PUT #1%, RECORD 6%, COUNT 512%\ZE.FLAG$=0%\GO TO 7000
14410 IF JSWIT%>=0% GO TO 32767
     14415 GOSUB 18000
14420 S$=CHR$(6$)+CHR$(8$)+CHR$(JOB$)+STRING$(24$,0$)+CHR$(255$)+STRING$(2$,0$)
14430 S$=SYS(S$)\GO TO 32767
                              CONVERT RSTS FORMAT DATE TO RT11 DATE
  1
15010 DAY#=VAL(LEFT(DATE*(DATE.RSTS*),2*))
15020 MONTH*=MID(DATE*(DATE.RSTS*),4*,3*)
15030 MONTH*=INSTR(1*,'JanrebMarAprHayJunJulAugSepOctNovDec',MONTH*)/3*+1*
15040 YEAR*=VAL(RIGHT(DATE*(DATE.RSTS*),6*))
15050 DATE.RT11*=YEAR*-72*+MONTH**1024*+DAY**32*
     15060 RETURN
    !
! CONVERT RT11 FORMAT DATE TO RSTS DATE 15080 YR = DAT. RT11 # AND 31 #
  15090 YR%=YR%+2%
15100 FOR I%=1% TO 366%
15110 DATE.RSTS%=YR%*1000+I%
    15120 GOSUB 15000
15130 IF DAT.RT11$=DATE.RT11$ GO TO 15160
15140 NEXT I$
15150 DATE.RSTS$==1$
15160 RETURN
  17000 1
                                 SQUEEZE DD: DEVICE
```

Computer Service Field Engineers

Join General Electric's expanding third-party computer service business.

Positions Available

Albany, NY • Metropolitan New York/New Jersey • Long Island • Philadelphia • Boston • Atlanta

The Challenge

Emphasizing quality people, quality service and quality multi-vendor products, General Electric is a recognized leader in the dynamically growing market for servicing computer equipment. These multi-location opportunities involve installing and maintaining various systems at customer sites.

Needed

You should have at least 3-5 years experience on DEC equipment, performing preventive and corrective hardware maintenance. Familiarity with third-party computer servicing, especially DEC-based, is highly desirable. VAX and/or PDP 11 training is required.

Rewards

General Electric offers an attractive salary with outstanding benefits, and all the advantages of a large and innovative corporation. To apply, please submit a resume and salary history, in confidence to:

Scott Hollander Ref. VXRSP Building 4 General Electric Co. 1 River Road Schenectady, NY 12345

The future is working at General Electric.



An Equal Opportunity Employer

CIRCLE 207 ON READER CARD

! !		
		å
17010	IF JSWIT\$ >0\$ THEN PRINT 'Detaching'	å
\	JSWIT#=-1#\S\$=SYS(CHR\$(6#)+CHR\$(7#)+CHR\$(128#))	•
	OPEN OUTDIR\$ FOR INPUT AS FILE #3%	
	INDEX\$ = 0\$	
	TYPE\$=2\$\SIZE\$=0\$\GOSUB 27000	
	IF SEG\$<0\$ GO TO 17070 M.PNT\$=PNT\$\INDEX\$=INDEX\$+1\$\GO TO 17040	
	CLOSE #3%\INDIR\$='[1,2]SQEZ'+NUM1\$(JOB\$)+'.CCH'	å
\	OPEN INDIR\$ FOR OUTPUT AS FILE #15	å
\	INREC%=1%\OUTREC%=2%	
	CLOSE #3\$	
	OPEN OUTDEV\$ AS FILE #2\$\CNT\$=0\$\INSIZE\$=M.PNT\$\SQ.FLAG\$=1\$\GO TO 7030	
17100	OPEN INDIR\$ FOR INPUT AS FILE #3% FOR I\$=0\$ TO 4\$	å
,	U\$(I\$)=DIR\$(O\$,I\$)	å
1		å
\	NEXT I\$\CLOSE #3\$OPEN INDIR\$ FOR INPUT AS FILE #3\$\text{	&
1	FOR I\$=0\$ TO 4\$	å
17110	DIR\$(0\$,1\$)=U\$(1\$) AND 255\$	
	NEXT I%\CLOSE #3% OPEN OUTDIR\$ FOR INPUT AS FILE #3%	
	FIRST.FILES\$=DIR\$(0\$,4\$)\JEX\$=12\$+DIR\$(0\$,3\$)/2\$	
17140	DIR\$(0\$,1\$)=0\$	&
\	DIR\$(0\$,2\$)=1\$	å
\	DIR\$(0\$,5\$)=512\$	&
,	DIR\$(0\$,6\$)=213\$	ě
'	DIR\$(0\$,7\$)=26385\$ DIR\$(0\$,8\$)=9852\$	Ł
,	DIR\$(0\$,9\$)=512\$-FIRST.FILES\$	ě
\ \	DIR\$(0\$,12\$)=0\$	&
\	DIR\$(0\$,JEX\$)=2048\$\CLOSE #3\$	
	INNAME \$='??????.???'	
17160	SQ.FLAG\$=-1\$\INDEV\$=INDIR\$\CLOSE #1\$,#2\$\GO TO 4190	å
1	•	å
i	ERROR OR COMPLETE MESSAGE OUTPUT	ě
!		å
19010	TE ISUTT# CO TO 18020	
	IF JSWIT GO TO 18030 PRINT OUT.MESS \$\return	
	CURKB\$=SYS(CHR\$(6\$)+CHR\$(9\$)+CHR\$(0\$))	å
1	CURKB\$=ASCII(MID(CURKB\$,2\$,1\$))\IF CURKB\$<128\$ GO TO 18020	
18040	CURKB\$=((NOT CURKB\$)AND 255\$)/2\$	å
1	CURKB\$=CHR\$(6\$)+CHR\$(-5\$)+CHR\$(CURKB\$)+'Message from KIM Job: '	ě
	+NUM1\$(JOB\$)+CHR\$(13\$)	ě
\	+CHR\$(10\$)+OUT.MESS\$+STRING\$(10\$,7\$)+CHR\$(13\$)+CHR\$(10\$) S\$=SYS(CURKB\$)\RETURN	å
20000		å
1		å
I	GET DD SUBROUTINE	å
1		ě
20010	FIELD #2\$,512\$ AS BUFIN\$\HUNG\$=0\$	
	LSET BUFIN\$=STRING\$(512%,0%)	
	GET #2%, RECORD REC%, COUNT 512%\RETURN	
24000		&
!	DAD DOMODY I ACTIVO	ě
!	DIRECTORY LISTING	ě
1		å
	DIR.FLAG\$=-1\$	å
1	IF INPS=05 THEN INDEV\$=OUTDEV\$\OUTDEV\$='KB:'	å
\	INPPN\$=OUTPPN\$\OUTPPN\$=''	å
\	INN AME \$= OU TN AME \$\OU TN AME \$= ' '	
200		ė.
\	INPP.WILD%=OUTPP.WILD%\OUTPP.WILD%=0%	&
1	INPP.WILD\$=OUTPP.WILD\$\OUTPP.WILD\$=O\$ INFIL.WILD\$=OUTFIL.WILD\$\OUTFIL.WILD\$=O\$	
	INPP.WILD%=OUTPP.WILD%\OUTPP.WILD%=0%	& &
\	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INPIL WILDS-OUTFIL WILDS-OS INPFILES-OUTFILES-VB: INPIS-OUTFILES-VB: INPIS-OUTFILES-VB: INPS-OUTFIS-OUTFIS-OUTPIS	& & &
\ \ \ 24020	<pre>INPP, WILD5=OUTPP.WILD5=O\$ INFIL.WILD5=OUTFILWILD5*OUTFILWILD5=O\$ INFILES=OUTFILE\$='UB:' INFIS=OUTFIS*OUTFILE\$='UB:' INFIS=OUTFIS*OUTFIS=CURPI\$ INFIS=OUTFIS*OUTFIS=CURPIS IF INSTR(15,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=CUTFILE\$+'.DIR'</pre>	& & &
\ \ \ 24020 24030	INPP, WILDS-OUTPP, WILDS-OSTPP, WILDS-OS INPFILES-OUTFILES-VOUTFILES-'KB:' INPFILES-OUTFILES-OUTFILES-'KB:' INPFILES-OUTFIS-CUPPIS INPES-OUTFIS-CUPPIS-CUPPIS IF INSTR(13,OUTFILES,'.')=05 THEN OUTFILES-CUTFILES+'.DIR' RETURN	& & &
\ \ 24020 24030 24040	<pre>INPP.WILD\$-OUTPP.WILD\$-O\$ INFIL.WILD\$-OUTPILWILD\$\OUTPILWILD\$-O\$ INFILE\$-OUTFILE\$-'KB:' INFI\$_=OUTFI\$_SOUTFIL=S-'KB:' INFI\$_=OUTFI\$_SOUTFI\$_=CURPI\$ INF2\$_=OUTF2\$_SOUTF2\$_CURP2\$ IF INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$</pre>	& & &
\ \ 24020 24030 24040	<pre>INPP,WILD\$=OUTPP.WILD\$=O\$ INFIL.WILD\$=OUTPILUD\$\OUTPILWILD\$=O\$ INFILE\$=OUTFILE\$=\UNID\$=O\$ INPIS=OUTPI\$\UNIPI\$=CURPI\$ INPIS=OUTPI\$\UNIPI\$=CURPI\$ INPIS=OUTPI\$\UNIPI\$=CURPI\$ IF INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=CUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE #1\$ OPEN INDIR\$ FOR INPUT AS FILE #3\$</pre>	& & & &
24020 24030 24040	<pre>INPP,WILD\$=OUTPP,WILD\$=O\$ INFIL.WILD\$=OUTPILWILD\$\OUTPILWILD\$=O\$ INFILE\$=OUTFILE\$=\UNID\$=O\$ INFILE\$=OUTFILE\$=\UNID\$=O\$ INFILE\$=OUTFILE\$=\UNID\$=O\$ INFICE\$=OUTFIS\UNIP\$=CURPI\$ INFIR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$2\$ ITFE\$=O\$ ITFE\$=O\$ ITFINAME\$<'' THEN F\$=INNAME\$\TYPE\$=4\$</pre>	& & & &
24020 24030 24040 \ \	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INPFILE\$-OUTPILE\$-WILDS-OUTPIL WILDS-OS INPFILE\$-OUTPIS-CUPPIS INPS-OUTPIS-VOUTPIS-CUPPIS INPS-OUTPIS-VOUTPIS-CUPPIS IPS -OUTPIS-VOUTPIS-CUPPIS IF INSTR(15, OUTFILE\$,'.')=0\$ THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ ITFES-OS IF INNAME\$<'' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$=0\$	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 \ \ \ 24050 24060	INPP.WILDS=OUTPP.WILDS+OUTPP.WILDS=O\$ INPIL.WILDS+OUTPILWILDS*OUTPILWILDS=O\$ INPILES=OUTPILES+'UB:' INPIS=OUTPIS*OUTPIS=CURPIS INP2S=OUTPES*OUTPIS=CURPIS IF INSTR(1%,OUTFILES,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ TYPES=O\$ IF INNAME\$<'' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX=O\$ INDIR\$ FOR INPUT AS FILE \$3\$	& & & &
24020 24030 24040 \ \ 24050 24060	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INFIL. WILDS-OUTPIL WILDS-OUTPIL WILDS-OS INFILES-OUTFILES-'RB:' INPIS-OUTPIS WITPIS-CURPIS INP2S-OUTPS WOUTPIS-CURPIS IF INSTR(15,OUTFILES,'.')=05 THEN OUTFILES-OUTFILES+'.DIR' RETURN OPEN OUTFILES FOR OUTPUT AS FILE #15 OPEN INDIRS FOR INPUT AS FILE #35 ITPSS-OS IF INNAMES<'' THEN F\$=INNAME\$\TYPES=45 INDEXS-OS MESSS=HDCRT.HEAD\$(15),4735,245) IF BLANK\$(15,2355)=29580 THEN FORMS='RSTS/E' ELSE FORM\$='RT11'	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 \ \ 24050 24060	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INFIL. WILDS-OUTPIL WILDS-OUTPIL WILDS-OS INFILES-OUTFILES-'RB:' INPIS-OUTPIS WITPIS-CURPIS INP2S-OUTPS WOUTPIS-CURPIS IF INSTR(15,OUTFILES,'.')=05 THEN OUTFILES-OUTFILES+'.DIR' RETURN OPEN OUTFILES FOR OUTPUT AS FILE #15 OPEN INDIRS FOR INPUT AS FILE #35 ITPSS-OS IF INNAMES<'' THEN F\$=INNAME\$\TYPES=45 INDEXS-OS MESSS=HDCRT.HEAD\$(15),4735,245) IF BLANK\$(15,2355)=29580 THEN FORMS='RSTS/E' ELSE FORM\$='RT11'	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 \ \ 24050 24060	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INFIL. WILDS-OUTPIL WILDS-OUTPIL WILDS-OS INFILES-OUTFILES-'RB:' INPIS-OUTPIS WITPIS-CURPIS INP2S-OUTPS WOUTPIS-CURPIS IF INSTR(15,OUTFILES,'.')=05 THEN OUTFILES-OUTFILES+'.DIR' RETURN OPEN OUTFILES FOR OUTPUT AS FILE #15 OPEN INDIRS FOR INPUT AS FILE #35 ITPSS-OS IF INNAMES<'' THEN F\$=INNAME\$\TYPES=45 INDEXS-OS MESSS=HDCRT.HEAD\$(15),4735,245) IF BLANK\$(15,2355)=29580 THEN FORMS='RSTS/E' ELSE FORM\$='RT11'	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 \ \ 24050 24060	<pre>INPP.WILD\$-OUTPP.WILD\$-OUTPP.WILD\$-O\$ INPIL.WILD\$-OUTPIL.WILD\$-OUTPIL.WILD\$-O\$ INPILE\$-OUTPILE\$-'WRIS' INPIS-OUTPIS-CURPY\$ INPIS-OUTPIS-CURPY\$ IP INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=CUTFILE\$+'.DIR' RETURN OPEN CUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN CUTFILE\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ IF INNAME\$<!--!' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$=O\$ MESS\$=MID(RT.HEAD\$(1\$),473\$,24\$) IF BLAKK\$(1\$,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(O\$,3\$)-->105 THEN FORM\$=FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(O) PRINT \$1\$, FORM\$;' FORM\$', 'FORM\$', 'DIR\$(O\$,0\$);' Directory segments '; DIR\$(O\$,2\$);' 'Open'</pre>	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 \ \ \ 24050 24060 \ 24065 24070	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INPFILES-OUTFILE WILDS-OUTPIL WILDS-OS INPFILES-OUTFILES-OUTPIS-CUPPS INPFILES-OUTPIS-OUTPIS-CUPPS INPS-OUTPIS-OUTPIS-CUPPS IF INSTR(15, OUTFILES, '.')-OS THEN OUTFILES-OUTFILES+'.DIR' RETURN OPEN OUTFILES FOR OUTPUT AS FILE #15 OPEN INDIRS FOR INPUT AS FILE #15 TYPES-OS IF INNAMES-O'' THEN FS-INNAMES-TYPES-45 IMPEXS-OS' MESSS-MID(RT.HEADS(15),4735,245) IF BLANKS(15,2355)-29580 THEN FORMS-'RSTS/E' ELSE FORMS-'RT11' IF DIRS(05,35)-105 THEN FORMS-FORMS+'+' PRINT #15, 'Directory for ';INDEV\$;' ';DATES(0) PRINT #15, FORMS: Format ';DIRS(05,05);' Directory segments '; DIRS(05,25);' Open' PRINT #15, WESSS	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 \ 24050 24060 \ 24065 24070	INPP, WILD\$-OUTPP, WILD\$-OUTPP, WILD\$-O\$ INPFILE\$-OUTPIS-CUPPT; INPFILE\$-OUTPIS-CUPPT\$ INPFILE\$-OUTPIS-CUPPT\$ INPS-OUTPIS-CUPPT\$ INPS-OUTPIS-CUPPT\$ INPS-OUTPIS-CUPPT\$ IPSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=CUTFILE\$+'.DIR' RETURN OPEN CUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IPSTR(0\$,0) IF INAME\$<\'' THEN F\$=INNAME\$\TYPP\$=\\$1\$ IPSTR(0\$,3\$)>105 THEN FORM\$=\RSTS/E' ELSE FORM\$=\RT11' IF DIR\$(0\$,3\$)>105 THEN FORM\$=\RTS/E' ELSE FORM\$=\RT11' IPDIR\$(0\$,3\$)>105 THEN FORM\$=\RTS/E' ELSE FORM\$=\RT11' IPDIR\$(0\$,2\$);' Open' INTI \$1\$, NESS\$ INTITUTE \$1\$, NESS\$ INTITUTE \$1\$, NESS\$ INTITUTE \$1\$ ACCOUNT DATE \$1\$ POS'	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 \ \ 24050 24060 \ 24065 24070	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INFIL. WILDS-OUTPIL WILDS-OUTPIL WILDS-OS INFILES-OUTFILES-'RB:' INPIS-OUTPIS-OUTPIS-CURPIS INPIS-OUTPIS-OUTPIS-CURPIS INPIS-OUTPIS-OUTPIS-CURPIS IF INSTR(15,OUTFILES,'.')=05 THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE #15 OPEN INDIR\$ FOR INPUT AS FILE #35 IF INNAME\$ IF INNAME\$ IF INNAME\$ IF INNAME\$ IF INNAME\$(05,05) IF BLANK\$(15,2355)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(05,35)>105 THEN FORM\$=FORM\$+'+' PRINT #15, FORM\$! FORM\$=FORM\$+'+' PRINT #15, FORM\$! FORM\$-'INDEV\$;'';DATE\$(0) PRINT #15, FORM\$! FORM\$-'INDEV\$;'';DATE\$(0) PRINT #15, MESS\$ HE\$='File Name Size Account Date Prot RTS Pos' PRINT #15, MESS\$	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 \ \ \ 24050 24060 \ 24065 24070 \ \ \ \ 24080 24080 24090	INPP.WILDS-OUTPP.WILDS-OUTPP.WILDS-O\$ INPIL.WILDS-OUTPIL.WILDS-O\$ INPILES-OUTPILES-'RB:' INPIS-OUTPIS-CURPI\$ INPIS-OUTPIS-CURPI\$ INPIS-OUTPIS-CURPI\$ INPIS-OUTPIS-CURPI\$ INFIR (15, OUTFILE\$,'.')=0\$ THEN OUTFILE\$=CUTFILE\$+'.DIR' RETURN OPEN CUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ IPF INAME\$ IF INNAME\$ IF INNAME\$ IF INAME\$ IF INAME\$ IF INAME\$ IF INAME\$ IF INAME\$ IF DIR\$(07,535)=05 MESS\$=MID(RT.HEAD\$(1\$),473\$,24\$) IF BLAKK\$(1\$,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(07,535)=05 IF DIR\$(07,535)=05 INTENT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, KESS\$ HES-FIRI Name Size Account Date Prot RTS Pos' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$ IF SEG\$ IF SEG\$ IN FERSASONO TO 500	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 24050 24060 24065 24070 1	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INPFILE\$-OUTPIS-OUTPIS-WILDS-OS INPFILE\$-OUTPIS-CUPPIS INPFILE\$-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS IF INSTR(15, OUTPIS-CUPPIS IF INSTR(15, OUTPIS-CUPPIS IF INSTR(15, OUTPIS-CUPPIS IF INSTR(15, OUTPIS-CUPPIS OPEN INDIR\$ FOR OUTPIS AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$<'' THEN \$5 = INNAME\$ \TYPE\$ = 4\$ INDEX\$-OS IF INNAME\$<'' THEN \$5 = INNAME\$ \TYPE\$ = 4\$ INDEX\$-OS MESS\$-MIDGR. HEAD\$(15),4735,245) IF BLANK\$(15,235) = 29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$)>10\$ THEN FORM\$=FORM\$+'+' PRINT \$1\$, FORM\$; Format ';DIR\$(0\$,0\$); DIrectory segments '; DIR\$(0\$,2\$); Open' PRINT \$1\$, MESS\$ HE\$=-File Name Size Account Date Prot RTS POS' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$ OTHER OF THEN DIR. FLAG\$=0\$\CLOSE \$1\$,\$3\$\GO TO 500 IF SEG\$ OTHER OF THEN DIR. FLAG\$=0\$\CLOSE \$1\$,\$3\$\GO TO 500	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 24050 24060 24065 24067 1	INPP.WILDS-OUTPP.WILDS-OUTPP.WILDS-O\$ INPIL.WILDS-OUTPIL.WILDS-OUTPILWILDS-O\$ INPILES-OUTPILES-URB'S INPIS-OUTPIS-CURPIS INPIS-OUTPIS-CURPIS INPIS-OUTPIS-CURPIS IF INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=CUTFILE\$+'.DIR' RETURN OFEN CUTFILE\$ FOR OUTPUT AS FILE \$1\$ OFEN INDIR\$ FOR INPUT AS FILE \$3\$ IF INAME\$!' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$=O\$ MESS\$=MID(RT.HEAD\$(1\$),473\$,24\$) IF BLAKK\$(1\$,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$) 10\$ THEN FORM\$-FORM\$+++' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, KESS\$ HE\$='File Name Size Account Date Prot RTS POS' PRINT \$1\$, HE\$ GOSUB 27000 T\$-DIR\$(0\$,0\$) THEN DIR.FLAG\$=0\$\CLOSE \$1\$,\$3\$\GO TO 500 T\$-DIR\$(0\$,0\$) THEN DIR.FLAG\$=0\$\CLOSE \$1\$,\$3\$\GO TO 500	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 (24050 24060 24065 24070 (24080 24100 241100 241100	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INPFILE\$-OUTPIS-OUTPIS-WILDS-OS INPFILE\$-OUTPIS-CUPPIS INPFILE\$-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS IF INSTR(15, OUTFILE\$,'.')=0\$ THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ IF INNAME\$<\'' THEN \$5 INNAME\$\TYPE\$=4\$ INDEX\$-05 IF INNAME\$<\'' THEN \$5 INNAME\$\TYPE\$=4\$ INDEX\$-05 MESS\$-HIDGR.HEAD\$(1\$),473\$,24\$) IF BLANK\$(1\$,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$)>10\$ THEN FORM\$=FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'FORM\$;' Format ';DIR\$(0\$,0\$);' Directory segments '; DIR\$(0\$,2\$);' Open' PRINT \$1\$, MESS\$ HE\$='File Name Size Account Date Prot RTS POS' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$<0\$ THEN DIR.FLAG\$=0\$\CLOSE \$1\$,\$3\$\GO TO 500 T\$=DIR\$(SST,ENTS-'\$) AND -256\$ T\$='CC ERROR >' T\$=ACT.FLI\$+' * 'IF T\$=256\$	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 (24050 24060 24065 24070 (24080 24100 241100 241100	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INPFILE\$-OUTPIS-OUTPIS-WILDS-OS INPFILE\$-OUTPIS-CUPPIS INPFILE\$-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS INPS-OUTPIS-CUPPIS IF INSTR(15, OUTFILE\$,'.')=0\$ THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ IF INNAME\$<\'' THEN \$5 INNAME\$\TYPE\$=4\$ INDEX\$-05 IF INNAME\$<\'' THEN \$5 INNAME\$\TYPE\$=4\$ INDEX\$-05 MESS\$-HIDGR.HEAD\$(1\$),473\$,24\$) IF BLANK\$(1\$,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$)>10\$ THEN FORM\$=FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'FORM\$;' Format ';DIR\$(0\$,0\$);' Directory segments '; DIR\$(0\$,2\$);' Open' PRINT \$1\$, MESS\$ HE\$='File Name Size Account Date Prot RTS POS' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$<0\$ THEN DIR.FLAG\$=0\$\CLOSE \$1\$,\$3\$\GO TO 500 T\$=DIR\$(SST,ENTS-'\$) AND -256\$ T\$='CC ERROR >' T\$=ACT.FLI\$+' * 'IF T\$=256\$	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	INPP.WILDS-OUTPP.WILDS-OUTPP.WILDS-OS INPIL.WILDS-OUTPIL.WILDS-OUTPILWILDS-OS INPILES-OUTPILWILDS-OUTPILWILDS-OS INPILES-OUTPILWILDS-OUTPIS-CURPIS INPIS-OUTPIS-CURPIS INPIS-OUTPIS-OUTPIS-CURPIS INPIS-OUTPIS-CURPIS IF INSTR(1%,OUTFILE\$,'.')=OS THEN OUTFILE\$=CUTFILE\$+'.DIR' RETURN OPEN CUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF SEANK\$(15,235)=S5 IF INAME\$<)' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$=O\$ MESS\$=MID(RT.HEAD\$(1\$),473\$,24\$) IF BLAKK\$(15,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$)>105 THEN FORM\$-FORM\$+++' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, FORM\$;' Format ';DIR\$(0\$,0\$);' Directory segments '; DIR\$(0\$,2\$):' Open' PRINT \$1\$, MESS\$ HE\$='File Name Size Account Date Prot RTS Pos' PRINT \$1\$, HE\$ GOSUB 27000 T\$-DIR\$(SEO\$,ENTX-1\$) AND -256\$ T\$-'< RENGR >' T\$-' <atf.hl\$+' '="" *="" if="" t\$="1024\$</td" t\$-act.fil\$+'=""><td>& & &</td></atf.hl\$+'>	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INFILES-OUTFILES-OUTFILES-'KB:' INPIS-OUTFILES-OUTFILES-CURPYS INPS-OUTFIS-CURPYS INPS-OUTFIS-CURPYS IF INSTR(1%,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=CUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN MOIR\$ FOR INPUT AS FILE \$1\$ IF INAME\$<\'' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$=O\$ MESS\$=HD(RT.HEAD\$(1%),473%,24%) IF BLAKK\$(1%,235%)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0%,3%)>105 THEN FORM\$=FORM\$*++' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'Porm\$,' Format ';DIR\$(0%,0%);' Directory segments '; DIR\$(0%,2%):' Open' PRINT \$1\$, MESS\$ HES='File Name Size Account Date Prot RTS Pos' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$<05 THEN DIR.FLAG\$=O\$\CLOSE \$1\$,\$3\$\GO TO 500 If SEG\$<05 THEN DIR.FLAG\$=O\$\CLOSE \$1\$,\$3\$\GO TO 500 If SEG\$<05 THEN DIR.FLAG\$=O\$\CLOSE \$1\$,\$3\$\GO TO 500 If SEG\$<05 THEN DIR.FLAG\$=O\$\CLOSE \$15,\$3\$\GO TO 500 If SEG\$<06 THEN DIR.FLAG\$=O\$\CLOSE \$15,\$3\$\GO TO 500 If SEG\$<07 THEN DIR.FLAG\$=O\$\CLOSE \$150 IF THEN DIR.FLAG\$ IF THEN DIR.FLAG\$\GO TO 500 IF SEG\$<07 THEN DIR.FLAG\$\GO TO 500 I	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INPFILE\$-OUTPIS-OUTPIS-UNIDS-OVTPI. WILDS-OS INPFILE\$-OUTPIS-CURPIS INPFILE\$-OUTPIS-CURPIS INPS-OUTPIS-CURPIS INPS-OUTPIS-CURPIS IF INSTR(15, OUTFILE\$,''.')=0\$ THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$-('' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$-05 IF INNAME\$-('' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$-05 IF BLANK\$(15,235)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF BLANK\$(15,235)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$)>10\$ THEN FORM\$=FORM\$+'+' PRINT \$1\$, FORM\$;' Format ';DIR\$(0\$,0\$);' Directory segments '; DIR\$(0\$,2\$);' Open' PRINT \$1\$, FORM\$;' Format ';DIR\$(0\$,0\$);' Directory segments '; DIR\$(0\$,2\$);' Open' PRINT \$1\$, MESS HE\$='File Name Size Account Date Prot RTS Pos' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$ IF SEG\$ OF THEN DIR.FLAG\$=0\$\CLOSE \$1\$,\$3\$\GO TO 500 T\$=DIR\$(SEG\$,ENT\$-1\$) AND -256\$ T\$=-1C ERROR >' T\$=-ACT.Fil.\$+' * 'IF T\$=-256\$ T\$=-4CT.Fil.\$+' 'IF T\$=-1024\$ IF DIR\$(0\$,3\$)<-10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(0\$,3\$)<-10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(0\$,3\$,4=10\$ GO TO 24150	& & & & & & & & & & & & & & & & & & &
240202 24030 24040 24050 24050 24065 24070 24100 24110 24120 24114 24124 24144 24144 24144	INPP, WILDS-OUTPL WILDS-OUTPL WILDS-OS INPFILE\$-OUTPIS-OUTPIS WILDS-OS INPFILE\$-OUTPIS-OUTPIS-CURPYS INPFIS-OUTPIS WILDS-OUTPIS INPS-OUTPIS WILDS-OUTPIS INPS-OUTPIS WILDS-OUTPIS INPS-OUTPIS WITPS-CURPYS IF INSTR(1%,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INAME\$ '' THEN F\$=INNAME\$\TYPPS=4\\$ INDEX\$=O\$ IF INNAME\$</'' THEN F\$=INNAME\$\TYPPS=4\\$ INDEX\$=O\$ MESS\$=MID(RT.HEAD\$(1\\$),\473\\$,24\\$) IF BLAKK\$(1\\$,2\\$) 10\\$ THEN FORM\$=*CRM\\$-'RSTS/E' ELSE FORM\\$-'RT11' IF DIR\$(0\\$,3\\$)>10\\$ THEN FORM\\$-CRM\\$-'', DATE\\$(0) PRINT \$1\\$, DORM\\$-' FOrmat ';DIR\\$(0\\$,0\\$)', Directory segments '; DIR\\$(0\\$,2\\$)', 'Open' PRINT \$1\\$, NESS\\$ IF SES\\$(0\\$) THEN BIR.FLAG\\$-0\\$\CLOSE \$1\\$,\$3\\$\QO TO 500 IF SEG\\$(0\\$) THEN DIR.FLAG\\$-0\\$\CLOSE \$1\\$,\$3\\$\QO TO 500 IF SEG\\$(0\\$, ENT\\$-1\\$) AND -256\\$ T\\$-'\\$ ERROR >' T\\$\\$-'\\$ ERROR >' T\\$\\$-\\$-\\$ ERROR >' T\\$\\$-\\$-\\$ IF T\\$-1024\\$ IF DIR\\$(SEG\\$, ENT\\$-1\\$) OF GO TO 24150 FOR J\\$-11\\$ TO 21\\$ IF DIR\\$(SEG\\$, ENT\\$+J\\$)\\$<0\\$ GO TO 24148 NEXT J\\$	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	INPP, WILDS-OUTPL WILDS-OUTPL WILDS-OS INPFILE\$-OUTPIS-CURPTY INPFILE\$-OUTPIS-CURPTY INPFILE\$-OUTPIS-CURPTY INPS-OUTPIS-CURPTY INPS-OUTPIS-CURPTY IF INSTR(1%,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1% OPEN INDIR\$ FOR INPUT AS FILE \$1% OPEN INDIR\$ FOR INPUT AS FILE \$1% IF INNAME\$ '' THEN F\$=INNAME\$\TYPE\$=4% INDEX\$=O\$ WESS\$=MID(RT.HEAD\$(1%),473%,24%) IF BLANK\$(1%,235%)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(O\$,3%) 105 THEN FORM\$=FORM\$+'+' PRINT \$1%, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1%, FORM\$;' FOrmat ';DIR\$(O\$,0%);' Directory segments '; DIR\$(O\$,28);' Open' PRINT \$1%, MESS\$ PRINT \$1%, MESS\$ PRINT \$1%, MES OSUB 27000 IF SEO\$ IF SEO\$ IF T\$=256\$ T\$='< ONUSED >'+SPACE\$(3%) IF T\$=512\$ T\$='< CRROR >' T\$=ACT.FIL\$+' ' IF T\$=1024% IF DIR\$(SEO\$,3%)<-SO OTO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(SEO\$,NIT\$+J\$)<>0\$ GO TO 24148 NEXT J\$ GO TO 24150 IF T\$=1024\$ THEN T\$=ACT.FIL\$+' A'	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILDS-OUTPP, WILDS-OUTPP, WILDS-OS INPIS-OUTPIS-UNIDS-OUTPIL WILDS-OS INPISE-OUTPIS-OUTPIS-CURPYS INPISE-OUTPIS WILDS-OUTPIS-CURPYS INPISE-OUTPIS WITPIS-CURPYS IF INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INDEX\$-O\$ IF INAME\$<\''' THEN F\$=INNAME\$\TYPPS=4\$ INDEX\$-O\$ MESS\$=MID(RT.HEAD\$(1\$),473\$,24\$) IF BLAKK\$(1\$,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$)>105 THEN FORM\$=FORM\$*'+' PRINT \$1\$, 'Directory for ';INDEV\$;''.DATE\$(0) PRINT \$1\$, 'Pormat'; Format';DIR\$(0\$,0\$);' Directory segments'; DIR\$(0\$,2\$);' Open' PRINT \$1\$, MESS\$ HE\$='File Name Size Account Date Prot RTS Pos' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$<0\$ THEN DIR.FLAG\$=O\$\CLOSE \$1\$, \$3\$\GO TO 500 T\$=DIR\$(SEG\$,ENT\$-1\$) AND -256\$ T\$='< (ENROR >' T\$=ACT.FIL\$+' * 'IF T\$=502\$ IF \$15*(UNUSED >++\$FACE\$(3\$) IF T\$=512\$ T\$=ACT.FIL\$+' * 'IF T\$=1024\$ IF DIR\$(SEG\$,ENT\$+1\$)<00 GO TO 24150 IF JIR\$(SEG\$,ENT\$+1\$)<00 GO TO 24150 IF JIR\$(SEG\$,ENT\$+1\$)</pre> GO TO 24150 IF T\$=1024\$ THEN TERM\$=1\$\text{VERR.LS}=LINE\GO TO 9110	& & & & & & & & & & & & & & & & & & &
\\\\24020 24030 24040 \\\\24050 24060 \\\\24060 \\\\24060 24100 241100 241120 24120 24124 24124 24144 24144 24144 24144 24144 24144 24144 24144 24150	<pre>INPP, WILD\$-OUTPP, WILD\$-OUTPP, WILD\$-O\$ INPFILE\$-OUTPIS-OUTPIL WILD\$-O\$ INPFILE\$-OUTPIS-CURPI\$ INPFILE\$-OUTPIS-CURPI\$ INPSOUTPIS-CURPI\$ INPSOUTPIS-CURPI\$ INPSOUTPIS-CURPI\$ INPSOUTPIS-CURPI\$ INPSOUTPIS-CURPI\$ INPSOUTPIS-CURPI\$ IN STR(1\$, OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ ITPES-O\$ IF INNAME\$<\'' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$-O\$ MESS\$-MIDGRT. HEAD\$(1\$),473\$,24\$) IF BLANK\$(1\$,235)=29580 THEN FORM\$+'RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$)>10\$ THEN FORM\$+FORM\$+'+' PRINT \$1\$, FORM\$; Format ';DIR\$(0\$,0\$); DIRECTORY SEGMENTS '; DIR\$(0\$,2\$); Open' PRINT \$1\$, FORM\$; Format ';DIR\$(0\$,0\$); DIRECTORY SEGMENTS '; DIR\$(0\$,2\$); Open' PRINT \$1\$, MESS\$ HE\$=!File Name Size Account Date Prot RTS POS' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$</pre> IF SEG\$ **T\$=ACT.FIL\$+' * IF T\$=256\$ T\$='< UNUSED >'+SPACE\$(3\$) IF T\$=512\$ T\$=ACT.FIL\$+' * IF T\$=1024\$ IF DIR\$(0\$,3\$)<-10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(0\$,3\$)<-10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(0\$,5\$)<-10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(0\$,5\$)<-10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(0\$,T\$)<-10\$ IF T\$=1024\$ THEN T\$=ACT.FIL\$+' A' IF T\$='< ERROR >' THEN ERM\$=1\$\text{VERR.L\$}=LINE\GO TO 9110 FILEX\$=5184\$ SEG\$\$,ENT\$-\$T\$-\$T\$.	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILDS-OUTPL WILDS\OUTPP.WILDS-O\$ INPFILE\$=OUTPIS OUTPIS WILDS\OUTPIS INPFILE\$=OUTPIS VOUTPIS CURPY\$ INPSECUTPIS VOUTPIS CURPY\$ INPSECUTPS VOUTPIS CURPY\$ IF INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$<!--'' THEN F\$=INNAME\$\TYPPS=4\$ INDEX\$=O\$ MESS\$=MID(RT.HEAD\$(1\$),473\$,24\$) IF BLAKK\$(1\$,2\$\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,3\$)-->105 THEN FORM\$=FORM\$*+'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'DORN\$;' Format ';DIR\$(0\$,0\$);' Directory segments '; DIR\$(0\$,2\$);' Open' PRINT \$1\$, NESS\$ HES='File Name Size Account Date Prot RTS PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$</pre> IF SEG\$ IF SEG\$ OF THEN DIR.FLAG\$=O\$\CLOSE \$1\$, \$3\$\QO TO 500 T\$=DIR\$(SEG\$,ENT\$-1\$) AND -256\$ T\$='< (ERROR >' T\$=*C\$ FORM\$=' IF T\$=1024\$ IF DIR\$(0\$,3\$)<=10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(SEG\$,ENT\$+J\$)<>0\$ GO TO 24148 NEXT J\$ GO TO 24150 IF T\$=1024\$ THEN T\$=ACT.FIL\$+' A' IF T\$=1024\$ THEN T\$=ACT.FIL\$+' A' IF T\$='< ERROR >' THEN ERM\$=1\$\LERRL\$=LINE\GO TO 9110 F.LEM\$=DIR\$(SEG\$,ENT\$-3\$), DATE.RSTS\$=''	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 24050 24060 24060 24070 24100 24110 24114 24114 24114 24114 24114 24114 24114 24114 24114 24117 24116	<pre>INPP, WILDS-OUTPL WILDS-OUTPL WILDS-OS INPFILE\$-OUTPIS-OUTPIS-CURPY INPFILE\$-OUTPIS-OUTPIS-CURPY INPFILE\$-OUTPIS-OUTPIS-CURPY INPS-OUTPIS-OUTPIS-CURPY IF INSTR(15, OUTFILE\$,'.')=O\$ THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$<'' THEN \$\frac{1}{2}\] IF INNAME\$<'' THEN \$\frac{1}{2}\] IF BLANK\$(15, 235) =29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF BLANK\$(15, 235) =29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF BLANK\$(15, 235) =29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$, 3\$)) = THEN FORM\$=FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, FORM\$;' Format ';DIR\$(0\$, 0\$);' Directory segments '; DIR\$(0\$, 2\$);' Open' PRINT \$1\$, 'MES\$ HE\$='File Name Size Account Date Prot RTS Pos' PRINT \$1\$, 'HE\$ GOSUB 27000 IF SEG\$</pre> IF SEG\$ IF SEG\$ ### OF THEN DIR.FLAG\$=0\$\CLOSE \$1\$, \$3\$\GO TO 500 ### SEG\$ ### OF T\$=ACT.FIL\$+' 'IF T\$=256\$ ### OLTP\$(0\$, 3\$, 0\$) = 10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(0\$, 3\$, 0\$=10\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(0\$, 2\$)*) >>> GO TO 24148 NEXT J\$ GO TO 24150 IF T\$=1024\$ THEN T\$=ACT.FIL\$+' A' IF T\$='< ERROR >' THEN ERRS'=1\$\close CERCE \close \close CERCE \clos	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 24050 24060 24060 24060 24120 24130 24142 24144 24150 24144 24140 24144 24140 24144 24140 24144 24150 24170	<pre>INPP, WILDS-OUTPP, WILDS-OUTP, WILDS-OS INPFILE\$-OUTPIS-OUTPIS-UNIDS-OVTIL WILDS-OS INPFILE\$-OUTPIS-OUTPIS-CURPYS INPS-OUTPIS-CURPYS INPS-OUTPIS-OUTPIS-CURPYS INPS-OUTPIS-CURPYS IF INSTR(15, OUTPISE\$,'.')=O\$ THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF BLANKS(15, 2355)=29580 THEN FORM\$-'RSTS/E' ELSE FORM\$-'RT11' IF DIR\$(0\$,3\$)>10\$ THEN FORM\$-FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$;'', 'DATE\$(0) PRINT \$1\$, 'DIrectory for ';INDEV\$;'', 'DATE\$(0) PRINT \$1\$, 'FORM\$;' FORMAT ';DIR\$(0\$,0\$);' Directory segments '; DIR\$(0\$,2\$);' Open' PRINT \$1\$, 'FORM\$;' FORMAT ';DIR\$(0\$,0\$);' DIRECTORY SEGMENTS' INFO THEN \$1\$, MESS INFE-FILE Name Size Account Date Prot RTS POS' PRINT \$1\$, 'HE\$ GOSUB 27000 IF SEG\$</pre> IF SEG\$ ### OF THEN DIR.FLAG\$-O\$\CLOSE \$1\$,\$3\$\GO TO 500 If SEG\$ ### OF THEN DIR.FLAG\$-O\$\CLOSE \$1\$,\$3\$\GO TO 500 If SEG\$ ### OF THEN DIR.FLAG\$-O\$\CLOSE \$1\$,\$3\$\GO TO 500 If SEG\$ ### OF THEN DIR.FLAG\$-O\$\CLOSE \$1\$,\$3\$\GO TO 500 If SEG\$ ### OF THEN DIR.FLAG\$-O\$\CLOSE \$1\$ #### OF THEN DIR.FLAG\$-O\$\CLOSE \$1\$ #### OF THEN DIR.FLAG\$-O\$\CLOSE \$1\$ #### OF THEN DIR.FLAG\$-O\$\CLOSE \$1\$ ##################################	& & & & & & & & & & & & & & & & & & &
24020 24030 24040 24050 24060 24060 24060 24120 24130 24142 24144 24150 24144 24140 24144 24140 24144 24140 24144 24150 24170	INPP, WILDS-OUTFP, WILDS-OUTFP, WILDS-OS INPFILE\$-OUTFIS-OUTFIS WILDS-OS INPFILE\$-OUTFIS-CURPYS INPFIS-OUTFIS-CURPYS INPS-OUTFIS-CURPYS INPS-OUTFIS-CURPYS IF INSTR(15, OUTFISE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ TYPE5-O\$ IF INNAME\$<\'' THEN F\$=INNAME\$\TYPE5=4\$ INDEX\$-O\$ MESS\$=MID(RT. HEAD\$(1\$),473\$,24\$) IF BLANK\$(1\$,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(O\$,3\$)>105 THEN FORM\$=FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(O) PRINT \$1\$, 'FORM\$;' FORMAT ';DIR\$(O\$,0\$);' Directory segments '; DIR\$(O\$,2\$);' Open' PRINT \$1\$, NES\$ HE\$='FILE Name Size Account Date Prot RTS POS' PRINT \$1\$, NES\$ GOSUB 27000 IF SEO\$ <td>& & &</td>	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILDS-OUTPL, WILDS-OUTPL, WILDS-OS INPFILE\$=OUTPIS-OUTFIL WILDS-OS INPFILE\$=OUTPIS-OUTPIS-CURPYS INPS-OUTPIS-OUTPIS-CURPYS INPS-OUTPIS-OUTPIS-CURPYS IF INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$<!--'' THEN F\$=INNAME\$\TYPPS=4\$ INDEX\$=O\$ IF INNAME\$</'' THEN F\$=INNAME\$\TYPPS=4\$ IF DIR\$(O\$,3\$)-->105 THEN FORM\$=FORM\$*-'RSTS/E' ELSE FORM\$='RT11' IF DIR\$(O\$,3\$)>105 THEN FORM\$=FORM\$*-'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(O) PRINT \$1\$, 'DORN\$;' Format ';DIR\$(O\$,0\$);' Directory segments '; DIR\$(O\$,2\$);' Open' PRINT \$1\$, NESS\$ HES='File Name Size Account Date Prot RTS POS' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$ IF SEG\$ IF SEG\$ IF T\$=125\$ T\$='< (INUSED) >+SPAGE\$ IF T\$=25\$ T\$='< (INUSED) >+SPAGE\$ IF T\$=122\$ IF DIR\$(SEG\$,ENT\$-1\$) AND -256\$ T\$='< (ERROR >' T\$='< INUSED) >+SPAGE\$ IF T\$=122\$ IF DIR\$(SEG\$,ENT\$+J\$)<>O\$ GO TO 24150 FOR J\$=11\$ TO 21\$ IF T\$=1024\$ THEN T\$=ACT.FIL\$+' A' IF T\$=102</pre>	& & & & & & & & & & & & & & & & & & &
\\\\24020 24030 24040 \\\\24050 24050 24060 \\\\\24060 24100 241100 241120 241140 241141 241141 241141 241141 241141 (241141 241141 (241141) (241141 (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (2411411) (241141) (241141) (241141) (241141) (241141) (241141) (241141	<pre>INPP, WILD\$-OUTPP, WILD\$-OUTPP, WILD\$-O\$ INPFILE\$-OUTPIS-OUTPIL WILD\$-O\$ INPFILE\$-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ IN SITE (15, OUTFILE\$,''.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPIT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ ITPES-O\$ IF INNAME\$<\'' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$-O\$ MESS\$-MIDGRT. HEAD\$(1\$),473\$,24\$) IF BLANK\$(1\$,235)=29580 THEN FORM\$+'+' PRINT \$1\$, FORM\$: FORM\$+'-' PRINT \$1\$, FORM\$-' PRINT</pre>	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILDS-OUTPP, WILDS-OUTP, WILDS-OS INPFILE\$-OUTPIS-OUTPIL WILDS-OS INPFILE\$-OUTPIS-OUTPIL WILDS-OS INPFILE\$-OUTPIS-CURPYS INPS-OUTPIS-OUTPIS-CURPYS INPS-OUTPIS-OUTPIS-CURPYS IF INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$<!--"--> IF DIR\$(O\$,3\$)>10,473\$,24\$) IF DIR\$(O\$,3\$)>105 THEN FORM\$=*CRM\$*-*** IF DIR\$(O\$,3\$)>105 THEN FORM\$=*CRM\$*-*** PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(O) PRINT \$1\$, 'DOPN' PRINT \$1\$, 'DOPN' PRINT \$1\$, 'DOPN' PRINT \$1\$, 'RE\$ GOSUB 27000 IF SEG\$</pre> IF SEG\$ IF S	& & & & & & & & & & & & & & & & & & &
\\\\24020 24030 24040 \\\\24050 24050 24060 \\\\\24060 24100 241100 241100 241140 241141 241141 241141 241141 241141 241141 241170 \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILD\$-OUTPP, WILD\$-OUTPP, WILD\$-O\$ INPFILE\$-OUTPIS-OUTPIL WILD\$-O\$ INPFILE\$-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ INPSI-OUTPIS-CURPI\$ IN SITE (15, OUTFILE\$,''.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPIT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$3\$ ITPES-O\$ IF INNAME\$<\'' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$-O\$ MESS\$-MIDGRT. HEAD\$(1\$),473\$,24\$) IF BLANK\$(1\$,235)=29580 THEN FORM\$+'+' PRINT \$1\$, FORM\$: FORM\$+'-' PRINT \$1\$, FORM\$-' PRINT</pre>	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILDS-OUTFP, WILDS-OUTFP, WILDS-OS INPFILE\$-OUTFILE\$-OUTFILE\$-VRLD\$-OS INPFILE\$-OUTFILE\$-OUTFILE\$-VRLD\$-OS INPFILE\$-OUTFILE\$-OUTFILE\$-VRLD\$-OS INPFILE\$-OUTFILE\$-OUTFILE\$-CURPY INPS-OUTFILE\$-OUTFILE\$-CURPY IF INSTR(1\$, OUTFILE\$,'.')-O\$ THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTFUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ INDEX\$-OS IF INNAME\$<-'' THEN \$\$\frac{1}{2}\] IF INNAME\$-'' THEN \$\$\frac{1}{2}\] IF BLANK\$(1\$, 235) = 29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$, 3\$).10\$ THEN FORM\$=FORM\$+'+' PRINT \$1\$, FORM\$;' Format ';DIR\$(0\$, 0\$);' Directory segments '; DIR\$(0\$, 2\$);' Open' PRINT \$1\$, FORM\$;' Format ';DIR\$(0\$, 0\$);' Directory segments '; DIR\$(0\$, 2\$);' Open' PRINT \$1\$, FORM\$;' Format ';DIR\$(0\$, 0\$);' Directory Segments '; DIR\$(0\$, 2\$);' Open' PRINT \$1\$, MESS HE\$='File Name Size Account Date Prot RTS Pos' PRINT \$1\$, HE\$ GOSUB 27000 If SEG\$(0\$ THEN DIR.FLAG\$=0\$\CLOSE \$1\$, \$3\$\GO TO 500 If\$-DIR\$(SEG\$, ENT\$-1\$) AND -256\$ If\$-CIPR\$(SEG\$, ENT\$-1\$) AND -256\$ If\$-CIPR\$(SEG\$, ENT\$-1\$) OF O O 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(SEG\$, ENT\$+J\$)\SO\$ GO TO 24148 NEXT J\$ GO TO 24150 IF IT\$-1024\$ THEN I\$=ACT.FIL\$+' A' IF IF\$=1024\$ THEN I\$=ACT.FIL\$+' A' IF IF\$=1024\$ THEN I\$=ACT.FIL\$+' A' IF IF\$=1024\$ THEN I\$=ACT.FIL\$+' A' IF T\$=-CERRO N' IF T\$=-1024\$ THEN I\$=ACT.FIL\$+' A' IF T\$</pre>	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILDS-OUTFP, WILDS-OUTFP, WILDS-OS INPFILE\$=OUTFILE\$\OUTFILE\$-\KID\$=OS INPFILE\$=OUTFILE\$\OUTFILE\$-\KID\$=OS INPFILE\$=OUTFILE\$\OUTFILE\$-\KID\$=OS INPFILE\$=OUTFILE\$\OUTFILE\$-\KID\$=OS INPFILE\$-OUTFILE\$\OUTFILE\$-\KID\$=OS INPFILE\$-OUTFILE\$,'.']=OS THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPIL AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$2\$ IF INNAME\$<\'' THEN F\$=INNAME\$\TYPE\$=4\$ INDEX\$=OS MESS\$=MID(RT. HEAD\$(1\$),473\$,24\$) IF BLANK\$(1\$,235\$)=29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(OS,3\$)>105 THEN FORM\$=FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$; ';DATE\$(O) PRINT \$1\$, 'FORM\$;'FORM**',FORM**', 'DATE\$(O) PRINT \$1\$, 'FORM\$;'FORM**', 'DER\$(O\$,0\$);' Directory segments '; DIR\$(O\$,28);' Open' PRINT \$1\$, 'RES\$ HE\$='File Name Size Account Date Prot RTS POS' PRINT \$1\$, 'RES\$ GOSUB 27000 IF SEO\$</pre> IF SEO\$ If SEO\$ **T\$=ACT.FIL\$** 'IF T\$=1024\$ T\$=ACT.FIL\$** 'IF T\$=1024\$ IF DIR\$(OS,3\$)<-105 GO TO 24150 FOR J\$=11\$ TO 21\$ IF DIR\$(SEO\$,AENT\$+J\$) **OO TO 24150 IF T\$=1024\$ THEN T\$=ACT.FIL\$** A' IF T\$='< ERROR 'THEN T\$=ACT.FIL\$** A' IF T\$='< ERRO	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILDS-OUTFL. WILDS-OUTFL. WILDS-OS INPFILE\$-OUTFIS-OUTFIL WILDS-OS INPFILE\$-OUTFIS-OUTFILE\$-'KB:' INPFIS-OUTFIS-OUTFIS-CURPYS INPFIS-OUTFIS-OUTFIS-CURPYS INPFIS-OUTFIS-OUTFIS-CURPYS IF INSTR(1\$,OUTFILE\$,'.')=O\$ THEN OUTFILE\$=OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTFUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$</pre> IF INNAME\$ IF DIR\$(0\$,3\$)>105 THEN FORM\$=\RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$,2\$):' Open' PRINT \$1\$, FORM\$:' Format ';DIR\$(0\$,0\$);' Directory segments '; DIR\$(0\$,2\$):' Open' PRINT \$1\$, FORM\$:' Format ';DIR\$(0\$,0\$);' DIRECTORY segments '; DIR\$(0\$,2\$):' Open' PRINT \$1\$, HE\$ GOSUB 27000 IF SEG\$ IF SEG\$ IF SEG\$ IF SEG\$ IF SEG\$ IF SEG\$ IF T\$=1024\$ IF DIR\$(SEG\$,ENT\$-1\$) AND -256\$ T\$='< ERROR >' T\$='< ERROR >' T\$=' <a' -="" vinuedd="">'+SPACE\$(3\$) IF T\$=512\$ T\$='<a' -="" vinuedd="">'+SPACE\$(3\$) IF T\$=512\$ IF DIR\$(SEG\$,ENT\$-3\$) IF DIR\$(SEG\$,ENT\$-3\$) OF OR J\$=11\$ TO 21\$ IF DIR\$(SEG\$,ENT\$-3\$) OF OR J\$=11\$ TO 21\$ IF DIR\$(SEG\$,ENT\$-3\$) OF OR J\$=15 TO 255\$ OF OR J\$=15 TO 24150 IF DATE,ENT\$-15 GO TO 24150 IF DATE,ENT\$</a'></a'>	& & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILD5-OUTFP, WILD5-OUTFP, WILD5-OS INPFILE\$-OUTFIS-OUTFILE\$-VRID\$-OS INPFILE\$-OUTFILE\$-OUTFILE\$-VRID\$-OS INPFILE\$-OUTFILE\$-OUTFIS-CURPYS INPSI-OUTFIS-OUTFIS-CURPYS INPSI-OUTFIS-OUTFIS-CURPYS IF INSTR(1%, OUTFILE\$, '.')-OS THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPIL AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$<\'' THEN \$\frac{1}{2}\] HEN \$\frac{1}{2}\] IF INNAME\$<\'' THEN \$\frac{1}{2}\] HEN \$\frac{1}{2}\] IF INNAME\$<\'' THEN \$\frac{1}{2}\] HEN \$\frac{1}{2}\] IF BLANK\$(1\$, 235) -29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$, 3\$)) -10\$ THEN FORM\$=FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'DIR\$(0\$, 3\$) -10\$ FIRE \$\frac{1}{2}\] INS\$-EFF1LE Name \$\frac{1}{2}\] INS\$-EF1LE Name \$\frac{1}{2}\] ACCOUNT\$, HEN DIR.FLAG\$=0\$\CLOSE \$1\$, \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\CLOSE \$1\$ \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\CLOSE \$1\$ \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\CLOSE \$1\$ \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\CLOSE \$15 \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\GO TO 24150 IF DIR\$\(0\$\) THEN DIR.FLAG\$=0\$\GO TO 24150 IF DIR\$\(0\$\) THEN DIR.FLAG\$=0\$\GO TO 24150 IF DIR\$\(0\$\) THEN THEN THEN THEN THEN THEN THEN THEN</pre>	\$\$6\$ & & & & & & & & & & & & & & & & & &
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<pre>INPP, WILD5-OUTFP, WILD5-OUTFP, WILD5-OS INPFILE\$-OUTFIS-OUTFILE\$-VRID\$-OS INPFILE\$-OUTFILE\$-OUTFILE\$-VRID\$-OS INPFILE\$-OUTFILE\$-OUTFIS-CURPYS INPSI-OUTFIS-OUTFIS-CURPYS INPSI-OUTFIS-OUTFIS-CURPYS IF INSTR(1%, OUTFILE\$, '.')-OS THEN OUTFILE\$-OUTFILE\$+'.DIR' RETURN OPEN OUTFILE\$ FOR OUTPIL AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ OPEN INDIR\$ FOR INPUT AS FILE \$1\$ IF INNAME\$<\'' THEN \$\frac{1}{2}\] HEN \$\frac{1}{2}\] IF INNAME\$<\'' THEN \$\frac{1}{2}\] HEN \$\frac{1}{2}\] IF INNAME\$<\'' THEN \$\frac{1}{2}\] HEN \$\frac{1}{2}\] IF BLANK\$(1\$, 235) -29580 THEN FORM\$='RSTS/E' ELSE FORM\$='RT11' IF DIR\$(0\$, 3\$)) -10\$ THEN FORM\$=FORM\$+'+' PRINT \$1\$, 'Directory for ';INDEV\$;' ';DATE\$(0) PRINT \$1\$, 'DIR\$(0\$, 3\$) -10\$ FIRE \$\frac{1}{2}\] INS\$-EFF1LE Name \$\frac{1}{2}\] INS\$-EF1LE Name \$\frac{1}{2}\] ACCOUNT\$, HEN DIR.FLAG\$=0\$\CLOSE \$1\$, \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\CLOSE \$1\$ \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\CLOSE \$1\$ \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\CLOSE \$1\$ \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\CLOSE \$15 \$3\$\GO TO 500 IF SEG\$<\(0\$\) THEN DIR.FLAG\$=0\$\GO TO 24150 IF DIR\$\(0\$\) THEN DIR.FLAG\$=0\$\GO TO 24150 IF DIR\$\(0\$\) THEN DIR.FLAG\$=0\$\GO TO 24150 IF DIR\$\(0\$\) THEN THEN THEN THEN THEN THEN THEN THEN</pre>	& & & & & & & & & & & & & & & & & & &

```
EX.SEG% IS THE SEGMENT TO BE EXTENDED
   25010 ENTRY.SIZE$=7$+(DIR$(0$,3$)/2$)
                     INDEX$=250$/ENTRY.SIZE$
TYPE$=0$\N$=0$\DIR.ERR$=0$\SEG$=EX.SEG$
GOSUB 27020
                     IF SEG$<>EX.SEG$ THEN ERM$=1$\ERR.L$=LINE\ GO TO 9110
  IF SEG$<>EX.SEG$ THEN ERM$=1$\ERR.L$=LINE\()
25020 GOSUB 25100
25030 DIR$(NEXT.SEG$,1$)=DIR$(SEG$,1$)
DIR$(SEG$,1$)=NEXT.SEG$+1$
DIR$(SEG$,1$)=NEXT.SEG$+1$
DIR$(SEXT.SEG$,0$)=DIR$(0$,0$)
25040 DIR$(NEXT.SEG$,0$)=DIR$(0$,3$)
DIR$(NEXT.SEG$,1$)=DIR$(0$,3$)
DIR$(NEXT.SEG$,1$)=DIR$(0$,3$)
DIR$(NEXT.SEG$,1$)=DIR$(0$,3$)
DIR$(NEXT.SEG$,1$)=DIR$(0$,3$)
DIR$(NEXT.SEG$,1$)=DIR$(0$,3$)
  25050 NEXT 1$
25060 DIR$(SEG$,ENT$-1$)=2048\RETURN
25100 IFIND UNUSED SEGMENT
25110 M$(I$)=0$ FOR I$=0$ TO 31$ \M$(I$)=-1$
  25110 M5(I$)=0$ FOR I$=0$ TO 31$ \M$(I$)=-1$
$$=0$
25112 NEXT.SEG$=DIR$(S$,1$)\IF NEXT.SEG$=0$ GO TO 25116
25114 M5(NEXT.SEG$==1$\S$=NEXT.SEG$-1$\OO TO 25112
25116 FOR I$=1$ TO DIR$(O$,0$)\IF M$(I$)=0$ GO TO 25150
25120 NEXT I$
25130 ERM$=4$\LERR.L$=LINE\GO TO 9110
25150 NEXT.SEO$=1$\L$\M$(I$)=-1$

FOR I$=1$ TO DIR$(O$,0$)\
DIR$(O$,0$)\LERT I$
  DIR$(0$,2$)=1$ IF M$(I$)
25160 NEXT I$
25170 RETURN
   27000 1
                    FIND A DIRECTORY ENTRY
          ENTRY PARAMETERS -- TYPE% =2 TO FIND AN EMPTY ENTRY
                                                                          TTPE$ =2 TO FIND AN EMPTY ENTRY
=4 FOR A PERMANENT FILE
=1 FOR A TENTATIVE FILE
=0 FOR ANY FILE OR EMPTY
SIZE$ THE SIZE IN BLOCKS OF EMPTY SPACE
INDEX$ ZERO FOR THE FIRST FILE MATCHING
(SIMILAR TO DIRECTORY SYS CALL)

F$ THE WILDCARD FILE NAME MUST BE
ALL 10 CHARACTERS XXXXXX.YYY
(SPACES INCLUDED IF REQUIRED)

SEG$ THE DIRECTORY SEGMENT NUMBER
STARTING FROM ZERO
(NEGATIVE IF ENTRY NOT FOUND)

NT$ THE ENTRY NUMBER WITHIN THE SEGMENT
STARTING FROM 1

NT$ THE START BLOCK OF THE FILE
.FIL$
           RETURN WITH
                                                             -- SEG%
                                                                      ENTS
PNT%
     32767 END
```



COM.BAS

By Alton O. Moore, III, Pharr, TX 78577

The following is a program which allows one to connect to a modem or to another computer connected to an RS-232 port on a RSTS system. The program will allow screen editing and will transfer files (stream) both ways. If you can access the file with a BASIC+ program, you can probably transfer it with this one.

To transfer files from my computer (the one running COM), I first log into the remote computer.

Type: PIP FILE.EXT = KB: -or- EDT FILE.EXT

Type: NULL/1T to ask COM to send the file (You are

prompted for the filespec)

When the file has been transferred, I usually type a †Z, which either tells PIP to close the file or tells EDT to change modes, at which point we exit EDT.

To copy files from the remote computer, we log, more or less, everything that computer sends to us to a specific file on our system. For example:

Type: PIP FILE.EXT (Don't hit < CR>)

Type: NULL/1R and enter the file we want to store

the info. in.

We are now back in interactive mode. Hitting a < CR> will cause the remote computer to execute our PIP command, copying the file we want to our screen. Everything you see on the screen also goes to the file you specified. When the file transfer is done (or when you've simply seen enough), typing another NULL/1R will close the output file. You will probably want to edit the beginning and end of the file just produced (I recommend using TECO for this) to remove prompts, command lines, empty lines, or whatever clutters up the data.

Note that while no files with attributes of any kind can be copied, you can convert these files to sequential files (usually with a record size of around 100), copy the sequential file, and rebuild the file that had attributes on the other computer. While we're on that subject, let me tell you the easiest way I've found to do this so far. Example:

We have an indexed file of record size 200. Use COBOL (I've never used BP2) to change the indexed file to a sequential file with a record size of 512 (we need to write out a block at a time). Let's say we set our 200 bytes of information (get rid of COMP fields!) into the leftmost 200 bytes of a block and write that block to the output file. Our FD should be described thusly:

FD OUTPUT-FILE

LABEL RECORDS ARE STANDARD

VALUE OF ID IS WS-FILE-ID.

O1 OUTPUT-REC PIC X(512).

Now we can open up this file with a BASIC+ program with a 512-byte buffer (OPEN "FILE.EXT" FOR INPUT AS FILE #1 /FIELD #1, 512 AS FILE.BUFFER\$). Every GET we execute gets us a block of the file. By extracting the leftmost 200 characters and writing them to another file, we finally have a stream format file. Since long strings have caused me nothing but problems in BASIC+, I would probably write pairs of 100-character records to the output file.

This program will copy such files with no problem. To reconstruct your file, reverse the procedure; or, you can skip a couple of steps by using IFL (an RMS utility) to reconstruct the indexed file from an appropriate sequential file.

You can hard-wire your computer to the remote computer if you wish. If you are going to be using dial-up modems there are special preparations. RSTS will no longer allow us to do I-O on a port set to modem-control without some rigamarole. Any modem or computer you want to communicate with should be plugged into a port without modem control. Some modems (such as the DFO3 series) will want to see DTR (pin 20) held high (which modem control on RSTS does) before they will allow dialing out or answer when hooked up to a non-modem port. On the DFO3 series, tying pin 9 to pin 20 will fix this. The solution should be similar for other modems.

The ports on both computers should be set to send and acknowledge XON and XOFF signals (on RSTS, set STALL and XON) to prevent loss of data.

```
! To end communications, type NULL/^C (NULL = CTRL/SPACE on a VT100).
! To send a file, type NULL/^T.
! To receive a file, type NULL/^R, and another when the transfer is complete.
! To receive a file, type NULL/^R, and another when the transfer is complete.
! To receive a file, type NULL/^R, and another when the transfer is complete.
! To receive a file, type NULL/^R, and another when the transfer is complete.
! To receive a file, type NULL/^R, and another when the transfer is complete.
! To receive a file, type NULL/^R, and another when the transfer is complete.
! Clear screen.
4 ON ERROR GOTO 20
5 OPEN "COM.DAT" FOR INPUT AS FILE #1 \INPUT #1, KB.ID$ \CLOSE #1 ! Get KB:
PRINT "Enter the KB: id and hir return when the modem carrier is on." &
PRINT "Keyboard id of modem (";KB.ID$;") "; \INPUT X$
!! IF X$
11 IF X$
12 ON ERROR GOTO 19000
13 GO TO 30
20 RESUME 21
21 KB.ID$="KB25:" \GOTO 8

30 OPEN "KB:" AS FILE #1, MODE 1+32 ! Open kb: in binary mode and allow ^S/^Q
40 FIELD #1, 128 AS KB.BUFFER$
50 OPEN KB.ID$ AS FILE #2, MODE 1+32
60 FIELD #2, 128 AS KB.BUFFER.2$
```

QUALITY SOFTWARE

for VAX/VMS and PDP-11/RSTS systems

REPORT WRITER

GRS is a powerful report writer and database query facility with English-like commands that can be used by non-programmers. It works with RMS and non-RMS files. A great productivity aid! (VAX and RSTS)

UTILITY

BACKUP/RESTORE DUMPER is a high-speed, easyto-use replacement for the \$BACKUP package (RSTS only).

DISK UTILITY PACKAGE

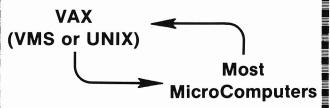
A set of utility programs which enable you to monitor disk allocation and free space as never before. An absolute MUST for well-structured disks! (RSTS only).



305 Madison Avenue New York, N.Y. 10165 (212) 972-1860 Telex: 177959 Cable: COMPUTERS

CIRCLE 4 ON READER CARD

FILE TRANSFER PROGRAMS



- Uses Christensen "MODEM" Protocol for compatibility WITH MOST MICROCOMPUTER-BASED COMMUNICATION PROGRAMS
- TRANSFERS ASCII FILES (WITH VARIABLE LENGTH RECORDS) AND/OR BINARY FILES (512 BYTE RECORDS)
- OPTIMIZED I/O USES MINIMAL VAX RESOURCES FOR EXTREMELY LOW SYSTEM LOADING
- CAN USE ALL STANDARD BAUD RATES BETWEEN 110 AND 9600
- Utility included to allow Multiple File Transfers
- VERY EASY TO USE

Low cost - \$500

WE ALSO HAVE COMPATIBLE COMMUNICATIONS PROGRAMS FOR RAINBOW, DEC MATE II, INTEL MDS, HEATH/ZENITH, ETC. FOR MORE INFORMATION, OR TO ORDER, CONTACT:

MicroProcessor Associates, Inc.

P.O. Box 3438 Nashua, NH 03061 (603) 888-9900 OR (617) 272-8086

```
90 LSET KB.BUFFER$=CHR$(10)+CHR$(13)+"Working...."+CHR$(10)+CHR$(13)
91 PUT #1, RECORD 4096, COUNT 15
120 ON ERROR GOTO 150 \GET #1, RECORD 8192, COUNT 1
120 OH SHRON GOID 150 GET #1, RECORD 0192, COURT 1
130 CHAR$-LEFT(KB. BUFFER$,1)
131 IF (LAST.CHAR$-CHR$(0)) AND (CHAR$-CHR$(3)) THEN 25000
132 IF (LAST.CHAR$-CHR$(0)) AND (CHAR$-CHR$(20)) THEN 26000
133 IF (LAST.CHAR$-CHR$(0)) AND (CHAR$-CHR$(18)) THEN 27000
                                                                                                     ! End....
                                                                                                     ! Sending....
! Receiving....
139 LAST.CHAR$=CHAR$
140 LSET KB.BUFFER.2$=CHAR$ \PUT #2, RECORD 4096, COUNT 1 \GOTO 120
150 RESUME 220
220 ON ERROR GOTO 250 \GET #2, RECORD 8192
                                                                           ! Get as many chars, as we can.
230 RECVD. DATA$=LEFT(KB. BUFFER. 2$, RECOUNT)
240 LSET KB. BUFFER$=RECVD. DATA$ \PUT #1, RECORD 4096, COUNT RECOUNT \GOTO 220
250 RESUME 110
19000 PRINT ERR, ERL \ON ERROR GOTO 0 \STOP
25000 PRINT \PRINT \PRINT "End of Communications Link...." \PRINT \GOTO 32767
26000 ! Section to transfer a file. The NULL/T causes this program to ask for
26066 GOTO 26060
                                                                            ! Go for another character
26070 ON ERROR GOTO 26080 \OPEN FILE.SPEC$ FOR INPUT AS FILE #3 \GOTO 26200 26080 RESUME 26090
26090 LSET KB.BUFFER$=CHR$(10)+CHR$(13)+"Illegal filespec"
26232 LSET KB. BUFFER$=RECVD. DATA$ \PUT #1, RECORD 4096, COUNT RECOUNT
                                                                                                      \GOTO 26230
26233 RESUME 26200
26240 IF ERR < 11 OR ERL <> 26200 THEN 19000
26250 RESUME 26260
26260 CLOSE #3 \LAST.CHAR$=""
             \LSET KB.BUFFER$=CHR$(10)+CHR$(13)+"File transferred"
\PUT #1, RECORD 4096, COUNT 18 \GOTO 110 ! Done
                                                                                      ! Done transferring file.
27000 ! Section to receive a file. ^R invokes this section. We ask for a 27010 ! filespec and open it for output. Interactive mode is reestablished 27020 ! so the user can invoke output from the other computer, which will 27030 ! be logged in the file until another null!^R is received...
27040 LSET KB.BUFFER$=CHR$(10)+CHR$(13)+"Enter filespec for logging: "
27050 PUT #1, RECORD 4096, COUNT 30 \FILE.SPEC$="" | Ask for and init...
27060 GET #1, COUNT 1 \IF LEFT(KB.BUFFER$,1)=CHR$(13) THEN 27070
27065 FILE.SPEC$=FILE.SPEC$+LEFT(KB.BUFFER$,1) \PUT #1, RECORD 4096, COUNT 1
27066 GOTO 27060
                                                                                        ! Ask for and init.....
27066 GOTO 27060 \ 1 Go for another character.
27070 ON ERROR GOTO 27080 \OPEN FILE.SPEC$ FOR OUTPUT AS FILE $3 \GOTO 27200
 27080 RESUME 26090
27230 CHAR$=LETT(KB.BUFFER$,1)
27233 IF (LAST.CHAR$=CHR$(0)) AND (CHAR$=CHR$(18)) THEN 27500
27239 LAST.CHAR$=CHAR$
27239 LAST. CHAR$=CHAR$
27240 LSET KB.BUFFER.2$=CHAR$ \PUT $2, RECORD 4096, COUNT 1 \GOTO 27220
27250 RESUME 27320
27320 ON ERROR GOTO 27350 \GET $2, RECORD 8192 ! Get as many chars. as we can.
27330 RECVD.DATA$=LEFT(KB.BUFFER.2$, RECOUNT) \GOSUB 27400
27340 LSET KB.BUFFER$=RECVD.DATA$

\PUT $1, RECORD 4096, COUNT RECOUNT

\GOTO 27320
27350 RESUME 27210
27400 BUTLD, BLOCK $= BUTLD, BLOCK $+ RECVD, DATA$
27401 IF LEN(BUILD.BLOCK$) < 512 THEN 27499
27410 LSET FILE.BUFFER$=LEFT(BUILD.BLOCK$,512)
```

27415 PUT #3 \BUILD.BLOCK\$ - DIDCE (BUILD.BLOCK\$,-512)
27415 PUT #3 \BUILD.BLOCK\$ - DIDCE (BUILD.BLOCK\$)
27416 IF LEN(BUILD.BLOCK\$) > 511 THEN PRINT "ERROR IN LINE 27416" \STOP

27500 ! Put the last block, close the file and we're done...
27510 LSET FILE.BUFFER\$=STRING\$(512,0) ! Fill buffer
27520 LSET FILE.BUFFER\$=BUILD.BLOCK\$! Set the day

27540 PUT #3 \CLOSE #3
27540 LSET KB.BUFFER\$=CHR\$(10)+CHR\$(13)+"Logging terminated"

\PUT #1, RECORD 4096, COUNT 20 \LAST.CHAR\$=""

32767 END

\GOTO 110

27499 RETURN

REPRINTS REPRINTS REPRINTS REPRINTS! All content in this publication is copyrighted

! Back to the usual.

All reprints must be purchased from M Systems, Inc. No other reprints are authorized.

All reprints shall contain both a

Price quotation available on request

! Fill buffer with nulls. ! Set the data into the buffer.

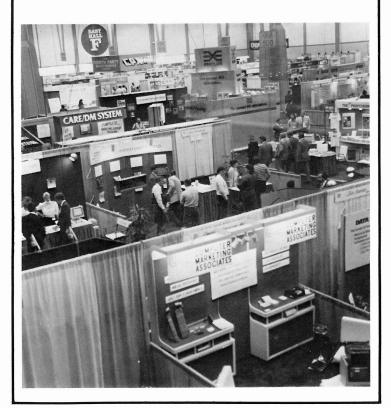
! Put last block and close.

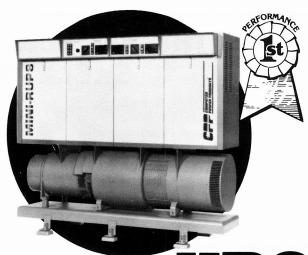
DEXPO



The fourth national DEC-compatible industry exposition, known as DEXPO opened on Sunday, October 23, 1983 at the Las Vegas Convention center. The exposition had a record 250 exhibitors covering about 48,000 square feet.

DEXPO has grown tremendously since the initial show and has doubled just since last year's DEXPO West.

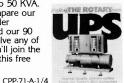




A WINNING COMBIN

Motor-generators provide the best power for sensitive computer systems. Add batteries and you've got RUPS, a price and a performance winner. Join

Add batteries and you've got RUPS, a price and a performance winner. Join the winner to protect your computer system from all utility power problems. There's single- or three-phase output from 12 to 50 KVA. And it doesn't even need air conditioning. Compare our Rotary UPS features with any UPS. Then consider performance, price, efficiency, maintenance and our 90 day unconditional money back guarantee to solve any of your power problems. If you do, we're sure you'll join the winner. Call toll free today (800) 421-6102 for this free brochure, and additional information.





227 E. Compton Blvd., Gardena, Ca 90248 (213) CPP-OWER (277-6937) (800) 421-6102 (Outside Ca) Tx: 6-74416

CIRCLE 212 ON READER CARD



survey of

150 DSKBLD users produced these overwhelming results:

100% of all respondents reported faster system response.

98% said it meets or exceeds expectations.

98% would recommend DSKBLD.

Here's why:

DSKBLD reduces overhead and file processor demands.

DSKBLD optimizes file clustersizes.

DSKBLD centers directories for faster seeks.

DSKBLD supports RSTS/E V8.0.

DSKBLD is easy to use and fully documented.

DSKBLD pays for itself within a few months. For complete information about DSKBLD's specifications, applications and unconditional money-back quarantee, call or mail your business card.



Pete Schnebele 1700 Westlake Ave. N. Seattle, WA 98109 206/285-3260

RSTS/E is a trademark of Digital Equipment Corp

AUTHORIZED digital COMPUTER DISTRIBUTOR

THE ALL SINGING, ALL DANCING SKELETON PROGRAM

By John D. Papadopoullos, Senior Analyst Programmer, K.K.N.G.H., Jeddah (K.S.A.)

Have you ever wished you could have RSTS/E specific internal information at your fingertips instead of having to write complex routines every time you have a bright idea?

Well your wish will come true with this all singing, all dancing skeleton program. I call it a skeleton program for your benefit because although the quick monitor access routines are there, they are not really used in this example.

The program is easily understood and modified. For example, a global change using any editor to the number 63 with the maximum jobs configured in your SIL will tailor the program to run on your system.

A word of warning: Although these routines have been tested at our installation, please take care to key them in correctly to avoid any corruption of monitor information. Strange things have been known to happen when you start experimenting at this level.

Below is an example of the basic results the program will produce in its current state:

```
DATE = 18-Jul-83
                                                                                                DATE = 18 Jul-83
JOB
              59,64
                                KB97
                                              NONAME
                                                                   TIME = 15:35
                                                                   TIME = 15:35
                                                                                                DATE = 18 Jul-83
       22
29
JOB
                                KB99
                                              NONAME
                                                                   TIME = 15:35
                                                                                                DATE = 18 \text{ Jul} - 83
              59,60
JOB
        30
               75,104
                                KB111
                                              NONAME
                                                                   TIME = 15:35
                                                                                                DATE = 18 \text{ Jul} - 83
                                                                                                DATE = 18 Jul-83
JOB
                                KB102
                                              NONAME
                                                                   TIME = 15:35
        36
                 1,34
                                                                   TIME = 15:35
TIME = 15:35
                                                                                                DATE = 18 Jul-83
DATE = 18 Jul-83
                 1,31
                                               NONAME
JOB
                                 KB98
                                               NONAME
                                                                   TIME = 15:35
                                                                                                DATE = 18 Jul-83
                1,222
              EXTEND
                 PROGRAM IDENTIFICATION BLOCK
                PROGRAM NAME
                                             MONIT.BAS
JOHN PAPADOPOULLOS
                                            JOHN PAPADOPOULLUS
6TH-MAR-1983
& KING KHALID NATIONAL GUARD HOSPITAL ( JEDDAH )&
THIS PROGRAM MILL ACCESS THE MONITOR TABLES
AND PRINT A READABLE REPORT TO AN OUTPUT FILE &
(MONIT.PRT) GIVING JOB STATE
JOB NUMBER
KEYBOARD NUMBER &
KEYBOARD NUMBER &
KEYBOARD NUMBER &

LOR NAME &
               KEYBOARD NUMBER
JOB NAME
TIME
DATE
THE PROGRAM WAS DESIGNED FOR EXPANSION WHICH EXPLAINS THE
EXCESS MONITOR INITIALIZATION ROUTINES AND COULD EASILY BE
MODIFIED TO USE THE VALUES IN THE TABLES FOR MANY RSTS/E
STATISTICS AND TUNING FUNCTIONS
               INFORMATION ON SPECIFIC PROGRAMS MAY BE PRINTED WITH A SMALL CHANGE AT LINE 3100 . S$ WILL CONTAIN THE PROGRAM NAME
                AT THE MOMENT ALL THE NONAMES WILL BE PRINTED
                TO TERMINATE THE PROGRAM ATTACH AND "Z THIS WILL CLOSE THE
               NO PATCHES ARE NECESSARY.
                THE CODE HAS BEEN TESTED BUT DONT BLAME ME IF IT DOESNT WORK
              ON ERROR GOTO 19000
INPUT "SLEEP PERIOD ";ZSLEEP
PRINT "Detachina...."
U$=$Y$(CHR$(6%)+CHR$(7%))
              SET STANDARD ERROR TRAP
INFORM USER OF SITUATION
DETACH PROGRAM
              OPEN "MONIT.PRT" FOR OUTPUT AS FILE 3%
GOSUB 1500
GOSUB 3000
SLEEP ZSLEEP
 100
```

```
GOTO 32767
                     GOTO 32767
                     DIM SYS.3%(63%)
DIM SYS.12%(63%)
                    DIM SY8.12%(63%)
DIM MX(83%)
DIM M2X(83%)
DIM Y1X(83%),Y2X(83%),Y3X(83%),Y4X(83%),Y5X(83%),Y6X(83%)
DIM Y7X(83%),Y8X(83%),Y9X(83%),Y0X(83%)
DIM Y7X(83%),Y8X(83%),Y9X(83%)
                                                                 INITIALIZATION
                     CHANGE SYS(CHR$(6%)+CHR$(-3%)) TO SYS.3%
                     SYS.3%(ZJOB.IND%)=SYS.3%(ZJOB.IND%)+SWAP%(SYS.3%(ZJOB.IND%+1%))
FOR ZJOB.IND% = 5% TO 29% STEP 2%
                     GET MONITOR TABLES PART 1
                     CHANGE SYS(CHR$(6%)+CHR$(-12%)) TO SYS.12%
 1520
                     SYS.12%(ZJOB.IND%)=SYS.12%(ZJOB.IND%)+
SWAP%(SYS.12%(ZJOB.IND%+1%))
FOR ZJOB.IND% = 3% TO 29% STEP 2%
                     GET MONITOR TABLES PART 2
1599
                                           MAIN PROGRAM LOGIC
 CNT.KB% = SYS.3%(3%)
MAXCNT% = SYS.3%(4%)
DEVCNT% = SYS.3%(5%)
DEVPTR% = SYS.3%(7%)
                                                                                         NUMBER OF KEYBOARDS
                                                                                         !MAXIMUM JOB NUMBER
!UNIT NUMBER TABLE
!DDB POINTER TABLE
                    DEVPTRX = SYS.3%(7%)
MEMLSTX = SYS.3%(6%)
JOBTBLX = SYS.3%(13%)
JESTSTX = SYS.3%(13%)
JEMAITX = SYS.3%(13%)
UNTCLUX = SYS.3%(15%)
UNTCLUX = SYS.3%(19%)
SATCTLX = SYS.3%(21%)
SATCTMX = SYS.3%(23%)
SATCTMX = SYS.3%(25%)
                                                                                        !MEMORY LIST ROOT
                                                                                       IMEMORY LIST ROOT

JOB TABLE

JOB STATUS TABLE

JOB WAIT FLAGS TABLE

IUNIT CLUSTERSIZE TABLE

IDISK STATUS TABLE

IFREE BLOCK COUNT TABLE

JOB STSTUS BITS TABLE

IFREE BLOCK COUNTS TABLE
                                                                                        !FREE SMALL BUFFER TABLE
!DEVICE NAME TABLE
!CSR TABLE
!NUMBER OF DISKS * 2
!HUNG TERMINAL COUNT
!JOB COUNT INFORMATION
!RUN TIME SYSTEM LIST
!ERROR LOG CONTROL TABLE
                    FREESX = $Y$.12X(3X)
DEVNAMX = $Y$.12X(5X)
CSRTBLZ = $Y$.12X(5X)
DECOMBX = $Y$.12X(5X)
TIYHCTX = $Y$.12X(13X)
JOBCNTX = $Y$.12X(13X)
FRLSLTX = $Y$.12X(13X)
SNDLSTX = $Y$.12X(13X)
DEUSYNX = $Y$.12X(21X)
DEUSYNX = $Y$.12X(23X)
DEUSYNX = $Y$.12X(23X)
DEUSYNX = $Y$.12X(23X)
CCLLSTX = $Y$.12X(27X)
3020
                                                                                        !ERROR LOG CONTROL TABLE !MESSAGE RECEIVER TABLE !LOGICAL NAME TABLE !DEVICE SYNONYM TABLE !PHYSICAL MEMORY SIZE !CCL TABLE
                     ZJOB.CNT% = 0%
ZJOB.IND% = 0%
                     CLEAR INDEX
                     ZJOB.CNT% = ZJOB.CNT% + 2%
ZJOB.TABLE% = PEEK(JOBTBL%+ZJOB.CNT%)
                     GOTO 3999 IF ZJOB.TABLE% = -1%
GOTO 3030 IF ZJOB.TABLE% = 0%
ZPOINT.JDB2% = PEEK(ZJOB.TABLE%+8%)
ZPOINT.DDB% = PEEK(PEEK(ZJOB.TABLE%))
ZPOINT.INT%=PEEK(ZPOINT.DDBX+30%) AND 255%
ZPOINT.JDPPN% = PEEK(ZPOINT.JDB2%+24%)
ZPROJECT.NUMBER% = SWAP%(ZPOINT.JDPPN%) AND 255%
ZPROGRAMMER.NUMBER% = ZPOINT.JDPPN% AND 255%
 3040
                      IS THIS THE END OF JOB TABLE
                     IS THIS THE END UF JOB TABLE
IS THIS AN ACTIVE JOB
ZPOINT. JOB2% = POINTER TO JDB2
ZPOINT. JOB2% = POINTER TO DDB
ZPOINT. INT% = POINTER TO THE INTERFACE TYPE
ZPOINT. JDPPN% = POINTER TO JDPPN
ZPROJECT.NUMBER% = PROJECT NUMBER
```

ZPROGRAMMER.NUMBER% = PROGRAMMER NUMBER IF (PEEK(ZPOINT.DDB%+2%) AND 255%)=ZJOB.CNT% AND (PEEK(ZPOINT.DDB%+6%) AND 8192%) <> 0% THEN ZDEVICE.STATE%=SWAP%(PEEK(ZPOINT.DDB%+2%)) AND 255% ELSE ZDEVICE.STATE% = -1% CHECK DEVICE ASSIGNMENT ZJOB.IND% = ZJOB.IND% + 1% Y1%(ZJOB.IND%) = ZJOB.CNT% Y2%(ZJOB.IND%) = ZJOB.TABLE% 3060 Y2%(2JOB.IND%) = ZJOB.TABLE% Y3%(ZJOB.IND%) = ZPOINT.JDB2% Y4%(ZJOB.IND%) = ZPOINT.JDB% Y5%(ZJOB.IND%) = ZPOINT.INT% Y6%(ZJOB.IND%) = ZPOINT.JDPPN% Y7%(ZJOB.IND%) = ZPROGRAMMER.NUMBER% Y9%(ZJOB.IND%) = ZPROGRAMMER.NUMBER% Y9%(ZJOB.IND%) = ZDEVICE.STATE% SAVE ALL THE MONITOR DATA IN THE ARRAYS FOR FURURE USE O\$=" JOB "+NUM1\$(ZJOB.CNT%/2%) IF ZPOINT.JDPPN% = O% THEN S\$="**,**" ELSE S\$=NUM1\$(ZPROJECT.NUMBER%)+","+NUM1\$(ZPROGRAMMER.NUMBER%) S\$=LEFT(S\$+" ",G%) 3070 DETERMINE JOB STATUS $$S$="(OPR)"$ IF ZPROJECT.NUMBER%=1% AND ZPROGRAMMER.NUMBER%=2% $$="(SELF)"$ IF ZPOINT.JDPPN%=P0% $$=""+S$ IF (ZJOB.CNT%/2%) < 10% 0=0$*+" "+S$ $$S$="KB"}$ IF ZPOINT.INT%=8% AND ZDEVICE.STATE% <> -1% THEN S\$="PK" ZDEVICE.STATE%=ZDEVICE.STATE%-P3% C%=(PEK (PEK (PZ%+ZDEVICE.STATE%+2%)+2%)AND 255%)/2% IF C% <> 0% THEN S\$="P"+NUM1*(ZDEVICE.STATE%)+"J" ZDEVICE.STATE% = C% CHECK FOR OPERATOR JOBS CHECK FOR SELF CHECK FOR PSEUDO KEYBOARDS S\$=S\$+NUM1\$(ZDEVICE.STATE%) \$\$\$="DET" IF ZDEVICE.STATE% = -1% 0\$=0\$+" "+S\$ \$\$=RAD\$(PEEK(ZPOINT.JDB2%+12%))+RAD\$(PEEK(ZPOINT.JDB2%+14%)) 0\$=0\$+" "+S\$ 3090 ZS\$=MID(0\$,22%,1%) IF S\$="NONAME" THEN GOSUB 4000 GOTO 3030 3100 3999 RETURN 4000! ** PRINT A LINE TO OUTPUT PRINT #3%.0\$,"TIME = ";TIME\$(0%);" ";"DATE = ";DATE\$(0%) 4030 4999 RETURN 19000!****** ERRORS WILL E N D U P ALL RESUME 3 IF ERR = 52 AND ERL = 3 CLOSE I% FOR I% = 1% TO 3% 19010



BUYING BIRTHDAY AND CHRISTMAS PRESENTS FOR MY HUSBAND ISN'T A PROBLEM ANYMORE!"

EMACS

FULL-SCREEN, MULTI-WINDOW EDITOR FOR UNIX AND VMS RUNS ON VAX AND MC68000

UniPress Software, Inc. is proud to announce the availability of the acclaimed Gosling EMACS on a variety of hardware configurations.

FEATURES:

- Full-screen, multi-window editor, allowing several files to be edited simultaneously.
- Compatible with EMACS on DEC 10/20.
- EDT emulation standard for VMS; optional for UNIX versions.
- Includes a terminal independent library allowing use of non DEC terminals.
- Direct access to system commands and user programs.
- Extensible through compiled MLISP and macros.
- Programming language syntax checking, indentation and spell checking.
- Debugging facilities.
- UNIX EMACS runs on Version 7, System III/V, Berkeley 4.1, 4.1C, 4.2.
- VMS EMACS runs on all currently supported DEC levels.

PACKAGING:

VAX/VMS and UNIX, MC68000/UNIX on Sun, Masscomp, Apollo, Tandy 16, Apple Lisa, Sritek Board for IBM PC, Dual, Plexus, Callan, and Cyb. Perg and Perkin Elmer, too. Inquire regarding other hardware.

PRICING:

Binary: from \$395. Source: from \$995.

OEM and dealer inquiries invited. Quantity terms available.

Call or write for more information

UniPress Software, Inc.

Dept. VRP

1164 Raritan Avenue, Highland Park, NJ 08904 Toll Free: 800-222-0550 (outside NJ) 201-985-8000 Telex: 709418

Mastercard and Visa

Overseas distribution available through Lifeboat Associates — Japan

Unix is a trademark of Bell Laboratories VMS is a trademark of Digital Equipment Corp.

Ready

VAX SYSTAT DOING IT THE RSTS WAY

By Bob Stanley, Computer Methods Corporation

As a recent RSTS to VAX convert, I haven't found many RSTS features that have not been replaced under VMS with something better or at least equivalent. Except one. . .SYSTAT. VMS has SHOW SYSTEM or SHOW PROC/CONTINUOUS but neither is as complete as the old familiar SYS/S. After searching long and hard for one status display that would tell me who was on the system, what terminal they were on, what image (program for all the RSTS people out there) they were running, what state the process (job) was in, the current CPU time, etc., I decided that nothing like that exists under VMS. So, being the kind of person who can't do without a good thing (and who recognizes the opportunity for a great article when he sees one!), I decided to attempt to write my own. SYSTAT.BAS was the result.

How It Works

SYSTAT uses the GET__JPI system service to extract information about each process running on the system (except for the NULL job and SWAPPER which it seems to ignore). GET__JPI is described in detail in the Guide to System Services. SYSTAT.BAS was written before the release of version 3.0 of VMS (and later converted to BASIC V2.0) which allows you to call GET__JPI directly from DCL. Thus SYSTAT is a BASIC V2.0 program as opposed to a DCL command procedure (which would have been much too slow anyway).

Executing SYSTAT

SYSTAT.BAS accepts certain switches via the LIB\$GET_FOREIGN run time library routine. In order to utilize this feature, a foreign command must be set up for executing SYSTAT. You can do this by placing the command

\$\$Y\$: = \$\delta\vertext{ci:[directory]}\$Y\$TAT.EXE in your login command file (any symbol will do if you don't like \$Y\$). You can then add switches to the symbol such as \$Y\$/\$Y\$TEM or \$Y\$/ACTIVE which will be used by \$Y\$TAT to determine what information to display.

The version of SYSTAT listed below will accept four switches. SYS/SYSTEM will display only the jobs whose associated UIC begins with "1,". SYS/USER will display all of the jobs whose UICs DO NOT start with "1,". SYS/ACTIVE will display all jobs that are currently associated with a keyboard (not detached).

There is a special switch /STAT that causes SYSTAT to provide some additional information such as working set sizes, working set quotas, page faults, etc. This can be handy if you need to tune the system for one reason or another.

The four switches can be combined in any way. If you wish to see the active users of the system just type SYS/ACTIVE/USER.

These four switches are just a sample of how you can alter SYSTAT to show any subset of the system you wish. If different user groups within your organization are divided by UICs then you can set up switches to identify different groups such as SYS/OPERATORS or SYS/PROGRAMMERS.

Restrictions

The version of SYSTAT listed below supports only VT100 terminals for the sake of simplicity. Add your own terminal types or add VT100 special effects simply by adding new "print using" masks. This version will reverse video any line of the display that pertains to the current user. It does this by translating a logical name called "USER_NAME" and comparing the result to the user name for each line of the display. This will only work, of course, if the logical name "USER_NAME" has been defined for any person who attempts to run SYSTAT. If you don't have this set up on your system, that part of the code should be removed. Or better yet, set up logicals for each user and then add a switch to SYSTAT that will allow you to say SYS/BOB or SYS/AL to get information about any particular user.

Conclusion

SYSTAT can be a handy little program to keep around for getting more complete system status displays than VMS normally gives. And besides, it's free, so what the heck, give it a try. Add your own features and see if the RSTS way isn't really better in this particular case.

Sample SYSTAT Display

Jobs Listed = 21

	VAX 11/780 sys	tem stat	cus as of 03	-Nov-83 at 02	:57	PM	
Username	Process Name	TT	Image	CPU Time	Pri	State	UIC
BOB	TTB4:	TTB4:		00:00:09.09	9	LEF	(100,1)
BOB	TTDO:	TTD0:	SYSTAT	00:00:54.25	4	CUR	(100,1)
CATHY	BSC\$XJB0		BSCPTPDET	00:00:05.38	6	LEF	(100,1)
BONNIE	BONNIE	TTA3:		00:02:20.24	4	COM	(100,2)
GARY	GARY	TTA1:		00:00:26.33	9	LEF	(100,6)
JEROL	JEROL	TTA4:	DOCUTL	00:05:03.81	4	LEF	(100,2)
BOB	BOB	TTC7:	S	01:28:03.39	4	COM	(100,1)
JOANNA	JOANNA	TTCO:	DISPMENU	00:00:58.57	7	LEF	(100,4)
SYSTEM	PC1	(5,5,5,5)	SPRINT	00:00:05.73	9	LEF	(100,1)
SYSTEM	PCO		SPRINT	00:00:14.61	7	LEF	(100,1)
SYSTEM	LP1		SPRINT	00:02:20.16	5	HIB	(100,1)
JOB CONTROL	PRTSYMB1		PRTSMB	00:00:01.67	9	HIB	(1,4)
SYSTEM	LCO		DOCL OG	00:00:25.14	8	LEF	(100,1)
SYSTEM	MSGMRG		MSGMRG	00:05:26.28	4	COM	(100,1)
SYSTEM	COMM_XY		MSGSWITCH	00:05:32.71	5	LEF	(100,1)
SYSTEM	SCHEDULE		SCHEDULER	00:00:02.83	9	LEF	(100,1)
SYSTEM	TDMS SWL		SWLPROC	00:00:00.18	9	HIB	(1,4)
SYSTEM	OPCOM		OPÇOM	00:00:00.33	10	LEF	(1,4)
JOB CONTROL	JOB_CONTROL		JOBCTL	00:00:12.18	11	HIB	(1,4)
SYSTEM	DRAOBACP		F11BACP	00:19:51.34	10	HIB	(1,3)
SYSTEM	ERRFMT		ERRFMT	00:00:01.95	12	HIB	(1,6)

```
DECLARE BYTE CONSTANT
                                                                    TRUE = (1 = 1), &
                                                                                                                                                                                                                                                                                                                         WORD JPI.FC.LENGTH, WORD JPI.FC, & LONG JPI.FC.LENGTH, & WORD JPI.FQ.LENGTH, WORD JPI.FQ, &
                                                                    FALSE = NOT TRUE, &
YES = TRUE, &
                                                                                                                                                                                                                                                                                                                        WORD JPI.FP.LENGTH, WORD JPI.FP. &
LONG JPI.FP.LENGTH, WORD JPI.FP. LENGTH, &
WORD JPI.FP.LOC,
LONG JPI.FF.LOC,
WORD JPI.VP.LENGTH, WORD JPI.VP, &
LONG JPI.FP.RET.LENGTH, &
WORD JPI.VP.LENGTH, WORD JPI.VP, &
                                                                    NO = NOT YES
                                                                                                                 JPI.WORKING.SET.QUOTA, & JPI.CURRENT.WORKING.SET, &
 DECLARE LONG
                                                                    JPI.CPU.TIME.
                                                                    JPI. PRIORITY.
                                                                                                                                                                                                                                                                                                                        WORD JPI.VP.LOC,
WORD JPI.WA.LENGTH,
LONG JPI.WA.LENGTH,
LONG JPI.WA.LENGTH,
LONG JPI.WA.LENGTH,
LONG JPI.WA.RET.LENGTH,
LONG JPI.WA.RET.LENGTH,
LONG JPI.WA.RET.LENGTH,
LONG JPI.WP.LENGTH,
LONG JPI.WP.RET.LENGTH,
&
                                                                                                                  JPI.UIC.MEMBER, &
JPI.DEFAULT.WORKING.SET, &
                                                                     JPI.UIC.GROUP,
                                                                   JPI.PID,
JPI.DIRECT.IO,
JPI.PAGE.FAULTS,
JPI.VIRTUAL.PEAK,
                                                                                                                 JPI.REMAIN.FILE.QUOTA, &
JPI.OPEN.FILE.QUOTA, &
JPI.MAX.WORKINC.SET, &
                                                                    JPI.OWNER,
                                                                                                                                                                                                                                                                                                                         WORD JPI.QW.LENGTH, WORD JPI.QW, &
LONG JPI.QW.LOC, LONG JPI.QW.RET.LENGTH, &
                                                                                                                  JPI.PEAK.WORKING.SET, &
                                                                    JPI.STATE,
                                                                                                                                                                                                                                                                                                                         WORD JPI.WS.LENGTH, WORD JPI.WS, &
LONG JPI.WS.LOC, LONG JPI.WS.RET.LENGTH, &
 EXTERNAL LONG FUNCTION
                                                                    OTS $CVT_L_TO, LIB $GET_FOREIGN, &
                                                                   SYS$GETJPI,
SYS$WAITFR,
                                                                                                       SYS$ASCTIM, &
LIB$SYS_TRNLOG, &
                                                                                                                                                                                                                                                                                                                          LONG DUMMY
                                                                                                                                                                                                                                                                                                                         STRING JPI. ADDRESS = 108% &
                                                                                                                                                                                                                                                      MAP (JPL_LIST)
                                                                    SYS &GETDEV
                                                                                                                                                                                                                                                                                                                         STRING JPI.TERMINAL = 7%, & STRING JPI.IMAGE.NAME = 64%, & STRING JPI.PROCESS.NAME = 15%, &
                                                                                                                                                                                                                                                      MAP (STRINGS)
 EXTERNAL INTEGER CONSTANT JPI$_USERNAME, JPI$_DFWSCNT, &
                                                                   JPI$_CPUTIM, JPI$_DIRIO, & JPI$_IMAGNAME, JPI$_FILCNT, &
                                                                                                                                                                                                                                                                                                                          STRING JPI.USERNAME = 12%
                                                                   JPI$ OWNER.
                                                                                                          JPI # FILLM. &
                                                                   JPI$_PRCNAM,
JPI$_PRI,
                                                                                                          JPI$_PAGEFLTS, &
JPI$_VIRTPEAK, &
                                                                                                                                                                                                                                   MAIN_PROGRAM:
                                                                                                                                                                                                                                                      LIB$ = LIB$$YS_TRNLOG ("USER_NAME",,CURRENT.USER$)
JOB.TYPE$ = O$
TYPE.SELECT$ = O$
                                                                   JPI$_STATE, JPI$_WSAUTH, & JPI$_TERMINAL, JPI$_WSPEAK, &
                                                                    JPI$ GRP.
                                                                                                          JPI$ WSQUOTA, &
                                                                                                                                                                                                                                                      L$ = LIB$GET_FOREIGN (SWITCH.STRING$,,SWITCH.LEN$)
GOSUB PARSE_SWITCHES UNLESS SWITCH.LEN$ = 0$
                                                                                                          JPI$_WSSIZE, &
SS$_NOMOREPROC, &
                                                                   JPI$_MEM,
                                                                                                                                                                                                                                                     JPI$ PID.
                                                                                                          SS$ BUFFEROVF, &
                                                                    SS$ NORMAL
                                                                    SS$_IVDEVNAM,
                                                                  BOLD = ESC + '[1m', &

REVERSE = ESC + '[7m', &

UNDERLINE = ESC + '[4m', &

BLINK = ESC + '[5m', &
DECLARE STRING CONSTANT
                                                                  (TYPE.SELECT% = 0% AND SWITCH.STRING$ = '') OR &
(TYPE.SELECT% = 0% AND SWITCH.STRING$ = PRINT.USER.NAME$) &
                                                                                                                                                                                                                                                                                               GOSUB PRINT_HEADINGS UNLESS HEADINGS.PRINTED$
GOSUB DISPLAY_STATISTICS IF DISPLAY.STAT$
GOSUB DISPLAY_JOB UNLESS DISPLAY.STAT$
                                                                                                                                                                                                                                                                           GOSUB GET_JPI_DATA
                                                                                                                                                                                                                                                      GOSUB DISPLAY_JOB_TOTAL
                                                                                                                                                                                                                                                      GOTO END_OF_PROGRAM
                                                                                                                                                                                                                                  GET_JPI_DATA:
                                                                                                                                                                                                                                                     E% = SYS$GETJPI (1% BY VALUE, &
                                                                                                                                                                                                                                                                                                 PIDADR BY REF,, &
JPI.ADDRESS BY REF,,,)
                                                                   RETURN
                                                                                                                                                                                                                                   PRINT HEADINGS:
                                                                                                                                                                                                                                                     WORD JPI.UN.LENGTH, WORD JPI.UN, &
LONG JPI.UN.LOC, LONG JPI.UN.RET.LENGTH, &
WORD JPI.CP.LOC, LONG JPI.CP.RET.LENGTH, &
LONG JPI.TM.LENGTH, WORD JPI.IN, &
LONG JPI.TM.LENGTH, WORD JPI.IN, &
LONG JPI.TM.LENGTH, LONG JPI.
 MAP (JPI_LIST)
                                                                   LONG JPI.OM.LENGTH, WORD JPI.OM, &
LONG JPI.OW.LENGTH, WORD JPI.OW, &
LONG JPI.OW.LOC, LONG JPI.OW.RET.LENGTH, &
                                                                   WORD JPI.PN.LENGTH, WORD JPI.PN, &
LONG JPI.PN.LOC. LONG JPI.PN.RET.LENGTH, &
                                                                   WORD JPI.PN.LDCO,
LONG JPI.PN.LDCO,
WORD JPI.PR.LENGTH, WORD JPI.PR. &
LONG JPI.PR.LDCO,
LONG JPI.PR.RET.LENGTH, &
LONG JPI.ST, &
LONG JPI.ST
                                                                                                                                                                                                                                                       PRINT
                                                                                                                                                                                                                                                       PRINT UNDERLINE; UNLESS DISPLAY.STAT$
                                                                                                                                                                                                                                                      PRINT ' Username Process Name TT
' Image CPU Time Pri State
UNLESS DISPLAY.STAT%
                                                                   WORD JPI.ST.LENGTH, WORD JPI.ST, & LONG JPI.ST.LOC, LONG JPI.ST.RET.LENGTH, &
                                                                                                                                                                                                                                                                                                                                                                          UIC
                                                                   WORD JPI.TT.LENGTH, WORD JPI.TT. & LONG JPI.TT.RET.LENGTH, & WORD JPI.UG.LENGTH, WORD JPI.UG. &
                                                                                                                                                                                                                                                                           Image Direct Opn Rmn ' + & 'Page Dflt Peak Max Wset Wset Crnt' & IF DISPLAY.STAT%
                                                                   LONG JPI.UG.LOC, LONG JPI.UG.RET.LENGTH, &
WORD JPI.UM.LENGTH, WORD JPI.UM, &
LONG JPI.UM.LOC, LONG JPI.UM.RET.LENGTH, &
                                                                                                                                                                                                                                                      PRINT UNDERLINE; IF DISPLAY.STAT$
PRINT Username Name I/O Cnt Opn ' + &
'Flts Wset Virt Wset Peak Quot Wset' &
                                                                   WORD JPI.UM.LEGO...,
LONG JPI.UM.LEGO...,
WORD JPI.PI.LENGTH, WORD JPI.PI. &
LONG JPI.PI.RET.LENGTH, &
LONG JPI.PI.RET.LENGTH, &
LONG JPI.DW, &
                                                                                                                                                                                                                                                                            IF DISPLAY.STAT%
                                                                                                                                                                                                                                                      PRINT RE.SET;
PRINT ESC + '[' + NUM1$(6%) + ';' + NUM1$(21%) + 'r'; &
UNLESS DISPLAY.STAT%
                                                                   LONG JPI.DW.LENGTH, WORD JPI.DW.LENGTH, WORD JPI.DW.LENGTH, WORD JPI.DW.LENGTH, & WORD JPI.DI.LENGTH, & LONG JPI.DI.LENGTH, & LONG JPI.DI.LENGTH, & LONG JPI.DI.RET.LENGTH, &
                                                                                                                                                                                                                                                       PRINT ESC + '[' + NUM1$(7%) + ';' + NUM1$(21%) + 'r'; &
```

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

```
IF DISPLAY.STAT%
                           PRINT ESC + '[' + NUM1$(6$) + ';' + NUM1$(1$) + 'H'; & UNLESS DISPLAY.STAT$
                                                                                                                                                                                                                                                                                                                                                                 PRINT USING STAT.MASK, PRINT.USER.NAME$, &
                                                                                                                                                                                                                                                                                                                                                                                                                                                            PRINT. IMAGE. NAME$, &
                           PRINT ESC + '[' + NUM1$(7$) + ';' + NUM1$(1$) + 'H'; & IF DISPLAY.STAT$
                                                                                                                                                                                                                                                                                                                                                                                                                                                            PRINT.DIRIO$, &
                                                                                                                                                                                                                                                                                                                                                                                                                                                            PRINT. OPEN$, &
PRINT. FILCNT$, &
                            HEADINGS. PRINTED% = TRUE
                                                                                                                                                                                                                                                                                                                                                                                                                                                            PRINT. PAGEFLTS$, & PRINT. DFWSCNT$, &
 PARSE SWITCHES:
                                                                                                                                                                                                                                                                                                                                                                                                                                                            PRINT. VIRTPEAK$,
PRINT. WSAUTH$, &
                          PRINT. WSPEAK$, & PRINT. WSQUOTA$, &
                                                                                                                                                                                                                                                                                                                                                                                                                                                            PRINT. WSSIZES
                                                                                                                                                                                                                                                                                                                                                                 TOTAL.OPEN.CNT% = TOTAL.OPEN.CNT% + (JPI.OPEN.FILE.QUOTA - & JPI.REMAIN.FILE.QUOTA)

TOTAL.WSSIZE% = TOTAL.WSSIZE% + JPI.CURRENT.WORKING.SET

JOB.TOT% = JOB.TOT% + 1%
                                                                                                                                                                                                                                                                                                                                                                  RETURN
                           INTERIOR (MITCH. STRINGS, DISPLAY. STRIPS)

IF DISPLAY. STRIT$

DISPLAY. TOTALS$ = INSTR(1$, SWITCH. STRING$, 1'/TOT')

SWITCH. STRING$ = LEFT(SWITCH. STRING$, DISPLAY. TOTALS$-1$) + &

RIGHT(SWITCH. STRING$, DISPLAY. TOTALS$+4$) &

IF DISPLAY. TOTALS$
                                                                                                                                                                                                                                                                                                                                       DISPLAY_JOB:
                                                                                                                                                                                                                                                                                                                                                                 PRINT REVERSE; IF CURRENT.USER$ = PRINT.USER.NAME$
PRINT USING PRINT.MASK, &
                                                                                                                                                                                                                                                                                                                                                                                                PRINT. USER. NAME$. &
                                                                                                                                                                                                                                                                                                                                                                                               PRINT.PROCESS.NAME$, &
EDIT$(LEFT(JPI.TERMINAL,6%),-1%), &
                             SWITCH.STRING$ = RIGHT(SWITCH.STRING$,2%) UNLESS SWITCH.STRING$ = ''
                            RETURN
                                                                                                                                                                                                                                                                                                                                                                                                PRINT. IMAGE. NAME$, & PRINT. CPU. TIME$, &
DETERMINE_JOB_TYPE:

JOB_TYPE$ = 1$ IF MID(PRINT.UIC$,2$,2$) = '1,'

JOB_TYPE$ = 2$ UNLESS MID(PRINT.UIC$,2$,2$) = '1,'

JOB_TYPE$ = 4$ UNLESS EDIT$(JPI.TERMINAL,-1$) = '
                                                                                                                                                                                                                                                                                                                                                                                                PRINT, PRIORITYS. &
                                                                                                                                                                                                                                                                                                                                                                                                 PRINT.STATE$, &
                                                                                                                                                                                                                                                                                                                                                                                               PRINT. UIC$
                                                                                                                                                                                                                                                                                                                                                                 JOB. TOT% = JOB. TOT% + 1%
PRINT RE.SET;
                                                                                                                                                                                                                                                                                                                                                                  RETURN
SETUP_JPI_VALUES:
                           JPI.UN.LENGTH = 12%
JPI.UN = JPI$_USERNAME
                                                                                                                                                                                                                                                                                                                                                               T_JPI_DATA:

PRINT.UIC.GROUP$ = FNLONG.OCT$(JPI.UIC.GROUP)

PRINT.UIC.MEMBER$ = FNLONG.OCT$(JPI.UIC.MEMBER)

PRINT.UIC$ = '(' + PRINT.UIC.GROUP$ + ',' + PRINT.UIC.MEMBER$ + ')'

GOSUB DETERMINE_JOB_TYPE

PRINT.PRIORITY$ = NUM1$(JPI.PRIORITY)

PRINT.STATE$ = FNCNV.STATE$(JPI.STATE)

IM.BRACKET$ = INSTR(IM.BRACKET$,JPI.IMAGE.NAME,')')

IM.DOT$ = INSTR(IM.BRACKET$,JPI.IMAGE.NAME,'')

PRINT.IMAGE.NAME$ = MID(JPI.MAGE.NAME,IM.BRACKET$+1$,IM.DOT$ - &

(IM.BRACKET$+1$))

CPU.TIME = (JPI.CPU.TIME / 6000) * 60

PRINT.CPU.TIME$ = FNONV.CPU.TIME$

PRINT.USER.NAME$ = EDIT$(LEFT(JPI.USERNAME,12$),-1$)

PRINT.PROCESS.NAME$ = EDIT$(LEFT(JPI.PROCESS.NAME,12$),-1$)

PRINT.PROCESS.NAME$ = EDIT$(LEFT(JPI.PROCESS.NAME,12$),-1$)
                                                                                                                                                                                                                                                                                                                                        EXTRACT_JPI_DATA:
                      JPI.UN.LENGTH = 12%
JPI.UN = JPI$_USERNAME
JPI.UN.LOC = LOC(JPI.USERNAME)
JPI.CP.LENGTH = 4%
JPI.CP.ENGTH = 4%
JPI.CP.CP.LENGTH = 64%
JPI.CP.LOC = LOC(JPI.CPU.TIME)
JPI.IM.LENGTH = 64%
JPI.IM.LENGTH = 64%
JPI.OM.LENGTH = 4%
JPI.OM.LENGTH = 15%
JPI.OM.LENGTH = 15%
JPI.OM.LENGTH = 15%
JPI.PN = JPI$_PRONAM
JPI.PN.LOC = LOC(JPI.PROCESS.NAME)
JPI.PN.LOC = LOC(JPI.PROCESS.NAME)
JPI.PR.LENGTH = 4%
JPI.PR.LENGTH = 4%
JPI.ST = JPI$_STATE
JPI.ST = JPI$_TERMINAL
JPI.TT = JPI$_TERMINAL
JPI.TT = JPI$_TERMINAL
JPI.TT = JPI$_TERMINAL
JPI.TI.LENGTH = 4%
JPI.TT = JPI$_TERMINAL
JPI.UI.LENGTH = 4%
                                                                                                                                                                                                                                                                                                                                    DISPLAY_JOB_TOTAL:

PRINT '-'; FOR I$ = 1$ TO 78$ IF DISPLAY.TOTALS$

PRINT '-' IF DISPLAY.TOTALS$

PRINT USING TOTAL.MASK, '',NUM1$(TOTAL.OPEN.CNT$),'','', &

NUM1$(TOTAL.WSSIZE$) &

IF DISPLAY.TOTALS$
                       JPI.TT.LOC = LOC(JPI.TERMINAL)
JPI.UI.LENGTH = 4%
JPI.UG = JPI$_GRP
JPI.UG_LOC = LOC(JPI.UIC.GROUP)
JPI.UG_LENGTH = 4%
JPI.UM = JPI$_MEM
JPI.UM.LOC = LOC(JPI.UIC.MEMBER)
JPI.UM.LOC = LOC(JPI.UIC.MEMBER)
JPI.UM.LOC = LOC(JPI.PID)
JPI.PI = JPI$_PID
JPI.PI.LOC = LOC(JPI.PID)
JPI.PI.LENGTH = 4%
JPI.DW = JPI$_DFWSCNT
JPI.DW = JPI$_DFWSCNT
JPI.DW.LENGTH = 4%
JPI.DW.LENGTH = 4%
JPI.DW.LENGTH = 4%
JPI.DI.UNC = LOC(JPI.DIRECT.IO)
                                                                                                                                                                                                                                                                                                                                                                  PRINT ESC + '[' + NUM1$(23%) + ';' + NUM1$(1%) + 'H' + BOLD;
                                                                                                                                                                                                                                                                                                                                                                PRINT DOUBLE. WIDE;
PRINT 'Jobs Listed = '; JOB.TOT%;
PRINT ESC + '[' + NUM1$(1$) + ';' + NUM1$(2\$) + 'r';
PRINT ESC + '[' + NUM1$(2\$) + ';' + NUM1$(1\$) + 'H'; RE.SET;
PRINT ESC + '[' + NUM1$(2\$) + ';' + NUM1$(1\$) + 'H'; RE.SET;
                                                                                                                                                                                                                                                                                                                                      DEF FNLONG.OCT$(OCT.VALUE%)
                                                                                                                                                                                                                                                                                                                                                                MAP (OCT_STRING) NEW.NUMBER$ = 4%
L$ = OT$$CVT_L_TO (OCT.VALUE$,NEW.NUMBER$)
OCT.NUMBER$ = NEW.NUMBER$
FNLONG.OCT$ = EDIT$(OCT.NUMBER$,-1%)
                        JPI.DI = JPI$_DIRIO
JPI.DI.LOC = LOC(JPI.DIRECT.IO)
JPI.DI.LENGTH = 4$
JPI.FC = JPI$_FILCNT
JPI.FC.LOC = LOC(JPI.REMAIN.FILE.QUOTA)
JPI.FC.LENGTH = 4$
JPI.FQ = JPI$_FILLM
JPI.FQ = JPI$_FILLM
JPI.FQ.LOC = LOC(JPI.OPEN.FILE.QUOTA)
JPI.FQ.LENGTH = 4$
JPI.PF = JPI$_PAGEFLTS
JPI.PF.LOC = LOC(JPI.PAGE.FAULTS)
JPI.PF.LENGTH = 4$
                                                                                                                                                                                                                                                                                                                                      DEF FNCNV.STATE$(CUR.STATE%)
                                                                                                                                                                                                                                                                                                                                                             CNV.STATE$(CUR.STATE$)

$ = 'CEF' IF CUR.STATE$ = 3$

$ $ = 'COLPG' IF CUR.STATE$ = 1$

$ $ = 'COM' IF CUR.STATE$ = 12$

$ = 'COM' IF CUR.STATE$ = 13$

$ $ = 'CUR' IF CUR.STATE$ = 14$

$ $ = 'HIB' IF CUR.STATE$ = 7$

$ = 'HIB' IF CUR.STATE$ = 5$

$ = 'LEFO' IF CUR.STATE$ = 5$

$ = 'LEFO' IF CUR.STATE$ = 6$

$ $ = 'LEFO' IF CUR.STATE$ = 6$

$ $ = 'MMAIT' IF CUR.STATE$ = 2$

$ $ = 'PWH' IF CUR.STATE$ = 2$

$ = 'PWH' IF CUR.STATE$ = 25$
                       JPI.PF.LOC = LOC(JPI.PAGE.FAULTS)
JPI.PF.LENGTH = 4%
JPI.VP = JPI$_VRITPEAK
JPI.VP.LOC = LOC(JPI.VIRTUAL.PEAK)
JPI.VP.LENGTH = 4%
JPI.WA.LENGTH = 4%
JPI.WA.LOC = LOC(JPI.MAX.WORKING.SET)
JPI.WA.LENGTH = 4%
JPI.WP.LOC = LOC(JPI.PEAK.WORKING.SET)
JPI.WP.LOC = LOC(JPI.PEAK.WORKING.SET)
JPI.WP.LOC = LOC(JPI.VORKING.SET.QUOTA)
JPI.QW.LOC = LOC(JPI.WORKING.SET.QUOTA)
JPI.QW.LENGTH = 4%
JPI.WS = JPI$_WSSIZE
JPI.WS.LOC = LOC(JPI.CURRENT.WORKING.SET)
JPI.WS.LENGTH = 4%
                                                                                                                                                                                                                                                                                                                                                               S$ = 'PFM' IF CUR.STATE$ = 4$

S$ = 'SUSP' IF CUR.STATE$ = 9$

S$ = 'SUSPO' IF CUR.STATE$ = 10$

FNCNV.STATE$ = S$
                                                                                                                                                                                                                                                                                                                                                                END DEF
                                                                                                                                                                                                                                                                                                                                     DEF FNCNV.CPU.TIME$
                                                                                                                                                                                                                                                                                                                                                              HR$ = CPU.TIME / 3600%
HR$ = NUM1$(HR$)
                                                                                                                                                                                                                                                                                                                                                             HR$ = NUM1$(HR$)

HR$ = '0' + HR$ UNLESS LEN(HR$) = 2$

MM$ = (CPU.TIME - (HR$ * 3600$)) / 60$

MM$ = NUM1$(MM$)

MM$ = '0' + MM$ UNLESS LEN(MM$) = 2$

SS = ((CPU.TIME - (HR$ * 3600$)) - (MM$ * 60))

SS$ = NUM1$(SS)

PER$ = INSTR(1$,SS$,'.')

SS$ = SS$ + '.00' IF PER$ = 0$

PER$ = INSTR(1$,SS$,'.') IF PER$ = 0$

SS$ = LEPT(SS$, PER$+2$)

SS$ = '00' + SS$ IF PER$ = 1$

SS$ = '0' + SS$ IF PER$ = 2$

SS$ = SS$ + '0' UNTIL LEN(SS$) = 5$

FNCNV.CPU.TIME$ = HR$ + ':' + MM$ + ':' + SS$
                          JPI.WS.LENGTH = 4%
DISPLAY_STATISTICS:
                        Y_STATISTICS:

PRINT.DFWSCNT$ = NUM1$(JPI.DEFAULT.WORKING.SET)

PRINT.DIRIO$ = NUM1$(JPI.DIRECT.IO)

PRINT.OPEN$ = NUM1$(JPI.OPEN.FILE.QUOTA - JPI.REMAIN.FILE.QUOTA)

PRINT.FILCNT$ = NUM1$(JPI.PEMAIN.FILE.QUOTA)

PRINT.PIECELIS$ = NUM1$(JPI.PEACE.FAULTS)

PRINT.VIRTPEAK$ = NUM1$(JPI.VIRTUAL.FEAK)

PRINT.WSAUTH$ = NUM1$(JPI.MAX.WORKING.SET)

PRINT.WSPEAK$ = NUM1$(JPI.WAR.WORKING.SET)

PRINT.WSQUOTA$ = NUM1$(JPI.WORKING.SET.QUOTA)

PRINT.WSSIZE$ = NUM1$(JPI.CURRENT.WORKING.SET)
                                                                                                                                                                                                                                                                                                                                                               END DEF
                                                                                                                                                                                                                                                                                                                                    END_OF_PROGRAM:
```

END

He's learning everything about the business.

READY FOR

Everything.

LOCK-11 is a system security and management package for RST\$.

LOCK-11 gives you absolute control of access by keyboard or user-I.D.

LOCK-11 provides an optional MENU environment that keeps non-privileged users where they belong.

LOCK-11 offers the system manager powerful surveillance utilities that actually improve thruput.

LOCK-11 is very well documented, supported and enhanced regularly.

LOCK-11 is available right now. Circle the response number below for a full set of documentation, or call 215-364-2800.



CIRCLE 12: ON READER CARD

NEW PRODUCTS



MULTIMODE MATRIX PRINTER NEW FROM DATASOUTH

Datasouth Computer Corporation has announced its DS220 multimode matrix printer. The DS220 provides correspondence printing for wordprocessing, draft quality printing for high speed data processing as well as dot-addressable graphics. Designed to operate with highperformance micro and mini-based business systems, the DS220 eliminates the need for multiple printers by combining these 3 different printing capabilities into a single unit. For further information, contact: Mr. James Busby, Datasouth Computer Corporation, 4216 Stuart Andrew Blvd., Charlotte, North Carolina 28210, (704) 523-8500, or Mr. Brian Cohen, Technology Solutions, Inc., 221 East 48th Street, New York, N.Y. 10017, (212) 752-6440.

EMULOGIC INTRODUCES VAX-BASED MULTI—USER SYSTEM

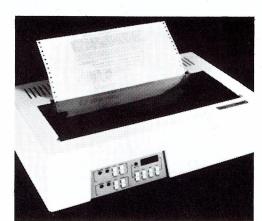
Emulogic, Inc. has introduced EMUNET™-2, the first highperformance, virtual disk multi-user microprocessor development system (MDS) that gives the full power of a VAX® operating system, EMUNET-2 supports a wide range of 8- and 16bit microprocessors and provides unique benefits not only to hardware and software designers but also to engineering project managers. The system can support as many as 60 widely dispersed hardware/software workstations (modified ECL-3211 standalone systems) at distances up to 5,000 feet from the host, and is the first multiuser MDS to provide true, highspeed 1-Megabaud data transfer between the host and workstations.

A typical EMUNET-2 system consisting of six satellite hardware/software workstations and six software-only stations, not including a customer-supplied VAX, is priced from \$110,000 to \$160,000. Additional hardware/software stations are priced between \$18,000 and \$25,000. Delivery is 30 to 60 days ARO with complete specifications. For further information contact Emulogic, Inc., Three Technology Way, Norwood, MA 02062-3978, (617) 329-1031, Telex 710-336-5908.

®Emulogic is a registered trademark of Emulogic, Inc. ®VAX, DEC, VMS, and VT 100 are all registered trademarks of Digital Equipment Corp. ™EMUNET is a trademark of Emulogic, Inc.

DECOLOR LA-120 PRINTER/COMPATIBLE WITH ISC 2400

Texprint announces a new addition to the family of DECOLOR LA-120 color printing upgrade modules. The DECOLOR/2400



Datasouth's new DS220 Multimode Matrix Printer has near letter quality printing.

provides plug-compatible color hard-copy printouts directly from the ISC 2400 color graphics display terminal.

Full compatibility with both ISC and DEC color graphics formats simplifies the task of obtaining fast, reliable color graphics in either "soft"- or "hard"-copy form. The new DECOLOR/2400 adds full-color graphics plotting on any LA-120, utilizing the character-graphics color display image data from an ISC 2400 terminal. DECOLOR/2400 combines character-graphics speed and economy with bit-mapped highsolution in the same unified system. The DECOLOR 2400/ISC 2400 package also provides convenient black/white or full-color printing for text and numerical data, creating a powerful color alphanumeric/graphics workstation capability.

Texprint's DECOLOR/2400 interface module can be installed in any LA-120 in less than one hour by any data terminal repair technician. The fully-assembled and tested DECOLOR upgrade kit, including handbook, interface cable, and one-year warranty, is available from stock directly from Texprint or independent terminal distributors at the single quantity price of \$1395. Quantity discounts are available. Contact Texprint, Inc., 8 Blanchard Rd., Burlington, MA 01803, or call (617) 273-3384 for more information.

NEW POWER LINE MONITORING DEVICE AVAILABLE FROM SOLA

A new power line monitoring device, featuring a printer that

ALL BACK ISSUES OF THE RSTS PROFESSIONAL



Includes a Holiday Special of 2 Library Cases

GREAT GIFT IDEA - ACT NOW - SUPPLY IS LIMITED

SIMPLY FILL IN COUPON BELOW AND SEND IT ALONG WITH YOUR CHECK TO: RSTS PROFESSIONAL, P.O. BOX 361, FORT WASHINGTON, PA 19034

□ Please rush my order for sets of ALL BACK ISSUES OF RSTS PROFESSIONAL @ \$100° per set. TOTAL ENCLOSED \$
☐ I have all the Back Issues but would like the Library Case(s) @ \$7.50 each. Send me Library Case(s). [Library Case holds 12 magazines.] TOTAL ENCLOSED \$
— DON'T FORGET YOUR FRIENDS AND BUSINESS ASSOCIATES —
NAME
ADDRESS
CITY/STATE/ZIP
PLEASE ENCLOSE YOUR CHECK WITH YOUR ORDER.

records all power problems diagnosed by the unit, has been introduced by Sola Electric, a unit of General Signal. This newest addition to Sola's power line monitor series prints out power interruptions and time of occurrence, enabling the user to determine the appropriate power conditioning equipment required. The unit also features a liquid crystal display that continuously indicates the time and allows for easy time/date setting.

Sola's Power Line Monitor/Printer is designed to provide accurate analysis of power line quality at potential electronic installation sites, or to check existing installations in order to determine whether electronics malfunctions are caused by incoming "dirty" power. It also offers a convenient reference chart of typical causes and recommended solutions for the most common power problems found on the line.

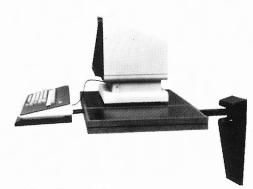
For more information on Sola's power line monitors, contact Sola Electric, 1717 Busse Road, Elk Grove Village, IL 60007. Telephone: 312/439-2800.

HIGH RELIABILITY LIGHT PEN NEW FROM INTERACTIVE

A new model light pen, the 250 SP, is now available from Interactive Computer Products, Inc. The pen features push-tip actuation, an electronic assembly with an MTBF in excess of 183,000 hours, and unsurpassed electro-optical performance. It is actuated by pressing the push-tip against the CRT over the light to be detected, requiring a force of less than 4 ounces. The polycarbonate push-tip actuator prevents marring of contacted surfaces. This highly reliable actuation mechanism has a life expectancy in excess of 2 million operations.

For more information, write Mr. Jim Bearden, Interactive Computer Products, Inc., 23591 El Toro Road, Suite 180, El Toro, California 92630. Phone: (714) 770-5332.

NEW WORKSTATION AIDS OFFICE EFFICIENCY



Adjustable table from Tumac Industries.

Tumac Industries, Inc. introduces Tabletech™, a versatile wall-mounted workstation with a wide range of applications. Its three easy-to-install components provide sturdy, efficient use with CRT displays, printers, copy machines, forms processing equipment, typewriters, acoustical printer enclosures, microfiche systems, and audiovisual equipment.

Available in a variety of surface sizes and laminate-hardwood finish combinations, Tabletech™ assures compatibility with every office environment. Tabletech™ components are sold separately with suggested list prices starting at \$59.95. Contact Tumac Industries, Inc., 650 Ford St., Colorado Springs, CO 80915-3798, (303) 596-4400, Telex 452-425 (Tumac Ind CSP).

INTELLIGENT SOFTWARE SYSTEM INTRODUCED BY FREY ASSOC.

Frey Associates, Inc. has introduced Themis^{™1}, a software system that may be the first practical solution to breaking the communication barrier between people and computers.

Themis understands questions and commands delivered via keyboard in plain English, and retrieves the requested information for output on a video screen or printer. Users can also ask Themis to perform a variety of functions such as sorting, logical comparisons, and calculations.

The first version of Themis runs on Digital Equipment Corporation's VAX-11™2 series of super minicomputers using the VAX/VMS™3 operating system. Themis is designed to run in 1.5 to 2 megabytes of main memory and requires either VAX-11 Datatrieve™4, or the Oracle™5 relational database management system.

Themis understands a basic vocabulary of more than 900 commonly-used English words, which can be easily expanded. Because Themis is so flexible, the system allows unskilled users to easily obtain information. Not only does Themis understand English sentences, sentence fragments, and slang exressions, but it also understands questions containing typos and misspelled words. Further information is available from Frey Associates, Inc., Chestnut Hill Rd., Amherst, NH 03031, (603) 472-5185.

Themis is a trademark of Frey Associates, Inc.

2/2/3'4VAX/11 and VAX/VMS, and Datatrieve are trademarks of Digital Equipment Corp.

™5Oracle is a trademark of Oracle Corporation.

jnet™ LINKS VAX/VMS AND IBM VM/SP USERS

Joiner Associates Inc. announces jnet™, a networking software product which allows VAX/VMS users to exchange files, electronic mail and real-time messages with IBM VM/CMS users. jnet enables peer-to-peer communication over bisynchronous lines using the RSCS (Remote Spooling Communications Subsystem) protocol.

Designed primarily for communication among interactive users, no special training is required to originate file transfers, mail and messages. Thus, non-technical users as well as programmers can benefit from the network. jnet provides easy-to-use commands for VAX/VMS users which correspond to CMS commands provided by

IBM. The IBM CMS commands are used unchanged to send information to users of VAX-11 nodes. Neither IBM nor VAX-11 users need to be concerned with learning unfamiliar command languages or file structures. jnet may also be used to send jobs to and receive output from batch processors attached to the RSCS network by Network Job Entry/Network Job Interface (NJE/NJI).

A perpetual license to jnet costs \$2500 — \$5000, depending upon the size of the VAX-11 processor. Special campus-wide licenses to four year colleges and universities cost \$250 until July 1, 1983 and \$2,000 thereafter. For more information about jnet, please call or write Joiner Associates Inc., JAnet Product Manager, 1124 Edgehill Drive, Madison, Wisconsin 53705. Telephone (608) 238-8134.

SYSTEM PERFORMANCE HOUSE ANNOUNCES UFD OPTIMIZER

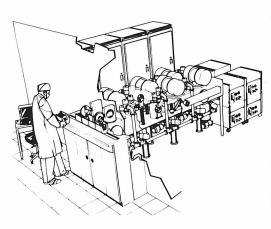
The System Performance House, Inc. is pleased to announce its latest high performance product for PDP-11s running RSTS/E. The UFD OPTIMIZER (UFDOPT) is available at the introductory price of \$250.00. The UFD OPTIMIZER is a fast REORDR-like utility which quickly and easily improves the performance of RSTS/E systems. It builds better structured directories which stay that way longer than those built by REORDR or RDR.

In addition to sorting on access or creation date in forward or reverse order, UFDOPT sorts the most important files in the directories to the front of the directory. A user modifiable table is supplied which defines the relative importance of files in the directory so that UFDOPT knows which files are most important at each site. For more information, write or call the System Performance House, Inc., 5522 Loch More Court, Dublin, Ohio 43017, 614-764-1585.

VARIAN OFFERS MULTISTATION SPUTTERING SYSTEM FOR DISKS

Varian Associates has introduced an automatic sputtering system designed specifically for processing 5¼ inch and smaller magnetic memory disks used by the rapidly expanding computer memory market.

Varian's new Model MDP-1000 System is a multistation cassette-tocassette sputtering system developed expressly for processing memory disks. It can clean and heat the substrates and sputter deposit the magnetic and nonmagnetic films required to produce thin film longitudinal mode rigid disks, as well as the vertical mode disks that may be widely used in the future. Initial units are scheduled for delivery in 1984. For further information write Varian Associates, Inc., 611 Hansen Way, Palo Alto, CA 94303.



An artist's rendering of Varian Associates' new MDP-1000 automatic sputtering system designed specifically for processing thin film magnetic memory disks.

EMULEX INTRODUCES SABRE NEW 51/4" DISK SUBSYSTEM FOR DEC LSI-11 COMPUTERS

Emulex Corporation has announced SABRE (Storage And Backup/RL02 Emulation), a packaged 51/4" Winchester disk subsystem which includes a high reliability 8" cartridge disk for backup and expanded storage capability. The new subsystem is software transparent and operates directly under all DEC LSI-11 operating systems.

The SABRE subsystem package consists of a 51/4" Winchester disk drive, an 8" cartridge disk drive, a power supply, and connecting cables; all of which are contained within a compact, attractive cabinet which mounts within a standard RETMA rack. The subsystem is functionally controlled by the Emulex UC01/LX emulating host adapter, which installs in the host CPU. The subsystem is fully compliant with FCC radiation emission standards for Class A equipment.

The SABRE subsystem has a list price of \$9595 per unit. OEM and quantity discounts are available. Delivery is thirty days ARO. The subsystem is the first in a completely new line of Emulex storage products that will take advantage of the versatile SCSI bus structure. For further information on the SABRE packaged subsystem, contact Flip Begich, Director of Domestic Sales, at Emulex Corporation, P.O. Box 6725, 3545 Harbor Blvd., Costa Mesa, CA 92626, (800) 854-7112. In California, (714) 662-5600.

INNOVATIVE ELECTRONICS OFFERS IBM 3270 BSC CONVERSION

Innovative Electronics, Inc. has announced the MC-80/600-1 communications processor. Emulating an IBM 3274-51C communications controller running configuration support level A, this device converts an inexpensive DEC VT-100 compatible terminal into a full function IBM 3277-1, 3277-2, 3278-1, 3278-2 terminal communicating with the IBM host using the EBCDIC binary synchronous communications (BSC) protocol.

Economical asychronous ASCII terminals such as the DEC VT-100 can be used in both local and dial-up applications. Full screen mapping

^{*}RSTS/E is a registered trademark of Digital Equipment Corporation; RDR is a registered trademark of Software Techniques, Inc.

is performed by the MC-80/600-1; data displayed on the asynchronous ASCII terminal will be the same as an IBM 3277/3278 display station, with virtual screen sizes 480, 960, and 1920 characters. All screen formatting capabilities are supported. For more information on the MC-80/600, please contact Diane Ruggiero, National Sales Manager at Innovative Electronics, Inc., 4714 N.W. 165th Street, Miami, Florida 33014; Telephone: (800) 327-3955; in Florida (305) 624-1644.

NEW TOOL ALLOWS SAFE PRINTED CIRCUIT BOARD REPAIR

Any company involved with printed circuit board repair (component replacement) can now minimize the discarding of difficult to repair and expensive PC boards using Nu-Concept's HART (Hot Air Repair Terminal). The HART machine is the most efficient individual component replacement tool for the repair of PC boards evidenced to date.

The unit is designed to desolder and resolder components on toughto-repair PC boards such as multilayer, aluminum, ceramic and steel boards. The machine operates on a directed-heat, controllable, hot air flow principle. Although HART is equally adapted to remove and replace all types of components, it is especially useful in the removing and replacing of the new chip carrier and gate array technology. The unit features low temperature operation to eliminate damage to PC boards and their components. Field service personnel like its compactness and portability. It is available from Nu-Concept Computer Systems, Inc., 1538 Bubbling Creek Road, Atlanta, GA 30319, 404-451-7737.

SOFPROTEX RELEASES DATASAFE TAPE LIBRARY SYSTEM

Sofprotex, a division of Government Copyright Services,

Page 90

has just released "DATASAFE", a tape library system which performs a wide variety of functions to aid the user in the restoration of lost or corrupt data. "DATASAFE" runs under RSTS/E.

DATASAFE is capable of maintaining from 1 to 10 separate computer system tape libraries and is currently used by several sites to keep track of PDP 11/70, and VAX tapes as well as data storage for other system configurations.

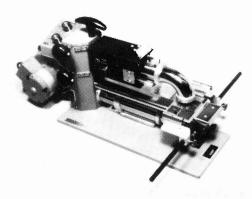
"DATASAFE" runs under the BASIC PLUS run-time system and keeps track of tape information utilizing a user-friendly interface. This menu-driven software product comes with full documentation and is easy to install.

SOFPROTEX supplies "DATASAFE" complete with software to generate tape labels for rack storage. For information call (415) 595-5595, or write to Sofprotex, P.O. Box 271, Belmont, Ca. 94002 U.S. A.

VOX OFFICE AUTOMATION NEW FROM AQUIDNECK

Aquidneck Data Corporation of Newport, RI, has announced a new entry into the Office Automation market — VOX, the Vax Office Exchange, an office automation system allows a Digital Equipment Corporation VAX computer user to exchange information between five different office application programs.

VOX is not just a menu driven collection of office programs. The system provides full transfer of



Nu-Concept's PC Board Repair Tool "HART"

information from one program to another, and entire files and documents from one user to another.

VOX was authored for non-technical business people to allow anyone in any office use of the computer systems. All programs included in VOX are integrated into one easy to follow menu that prompts the operator for successive commands. The program's advanced Send/Receive function allows transfer of files, documents and other data between people in the same office or across the country. For more information, contact Larry Wood, Aquidneck Data Corporation, (401) 847-7260, ext. 334.

CCA UNVEILS TEXT EDITOR FOR VAX UTILIZING UNIX OR VMS

Computer Corporation of America (CCA) recently introduced CCA EMACS, the EMACS text editor with the greatest combination of power, speed and functionality for Digital Equipment Corporation's VAX systems. CCA EMACS runs on DEC's complete line of VAX systems utilizing Berkeley UNIX or VMS, providing users with the greatest programming and text processing support yet for those systems. Support for CCA EMACS on other UNIX systems is planned. (UNIX is a trademark of Bell Laboratories.)

CCA EMACS is immediately available. It is competitively priced from \$350 to \$850, with volume discounts available and special rates for OEMs and educational institutions. Contact Computer Corporation of America for more information. CCA Headquarters is located at Four Cambridge Center, Cambridge, MA 02142, (617) 492-8860.

EMC CORPORATION OFFERS FIRST 2MBYTE MEMORY CARD FOR VAX-11/780

Following Digital Equipment Corporation's first shipments of the new VAX 11/780, EMC Corporation has announced a completely compatible 2MByte Add-in Memory Card. This high density memory card is particularly significant in that it is not only the first 2MB card, but also because all new VAX-11/780 CPUs are now required to be expanded in 2MB increments. The new VAX-11/780 will address up to 32MB.

Both this memory card, which is

designated the VX-2MB-780, and the depopulated 1MByte version, the VX-1MB-780, use 64K RAMS. They are functionally identical, including interleave, extended cycle, ECC and diagnostic compatibility with the DEC 1MB equivalent MS780-F(X) memory. The EMC Memory Card inserts directly into the DEC MS780E Memory System. No changes are required to the existing hardware

configuration or software.

The VX-2MB-780 and VX-1MB-780 are available for delivery from stock and bear a five year warranty. Single quantity price \$6,300 for 2MB and \$3,400 for 1MB. This is approximately 30% below DEC's equivalent price.

EMC Corporation is located at 385 Elliot Street, Newton, Mass. 02164. Telephone 1-(800)-222-EMC2.

PRODUCT UPDATES



ROSS SYSTEMS ANNOUNCES MAPS/AP PACKAGE

An integrated and interactive accounts payable package with extensive cash management features, designed specifically for use on the DEC VAX series of minicomputers, has just been announced by Ross Systems.

Using its full interactive capabilities, MAPS/AP handles sophisticated payables processing, check writing, vendor information, cash management, inquiries and reports.

Designed to integrate with all Ross software, MAPS/AP is particularly effective when used as a complement to the comprehensive general ledger/financial management package, MAPS/GL. It automatically updates payables to the general ledger and provides a GL distribution by multiple account structures, for example. In addition, MAPS/AP processing screens conform to the MAPS/GL format, reducing the time needed to become familiar with the system.

MAPS/AP, backed by a strong support team, will be available in January 1984 at a cost of \$15,000 — \$20,000 depending on VAX model. Ross Systems specializes in

integrated financial management software for DEC computers and is located at 1860 Embarcadero Road, Palo Alto, CA 94303. Tel: (415) 856-1100.

IMSL LIBRARY FOR VAX/UNIX

IMSL, Inc., an international firm specializing in technical and scientific software, has announced that the widely-used IMSL Library package is now compatible with VAX systems running the UNIX operating system.

The IMSL Library contains 540 FORTRAN subroutines, applicable to a broad range of mathematical and statistical functions. The Library is designed to reduce the time and cost involved in developing scientific and engineering application programs by allowing the programmer to select subroutines from this comprehensive library, rather than writing them.

IMSL is based at 7500 Bellaire Boulevard, Houston, Texas 77036. The company invites toll-free telephone inquires at (800) 222-IMSL. Texas residents may call (713) 772-1927. VERSION 4.0 OF TSX-PLUS NOW FROM S&H COMPUTERS

Version 4.0 of the TSX-Plus operating system, featuring increased virtual address space, improved user security, and compatibility with the latest version of the Digital Equipment's RT-11 operating system, has been announced by S&H Computer Systems, Inc.

The operating system, which allows RT-11 users to work in a multi-user, multi-tasking environment, has met with widespread acceptance among business, research, and program development users.

The new version will address up to 4Mb of main memory on PDP-11 systems that have either the extended UNIBUS, or the 22-bit Q-BUS together with the PDP-11/23 CPU. In addition, TSX-Plus 4.0 provides the RT-11XM environment, supporting program logical address space (PLAS).

TSX-Plus is listed at \$2,000, with an annual \$500 fee for support and the option to order upgrades. For further information, contact Gary Manookian or Richard Dohrmann, (615) 327-3670, or write to S&H

Computer Systems, Inc., 1027 17th Ave. South, Nashville, TN 37212.

EG&H RELEASES VSORT V3

Evans Griffiths and Hart, Inc. announced the release of Version 3 of VSORT, their very fast sort package running under VMS on Digital Equipment Corporation's VAX-11 computers. The new release sorts large files 20% to 50% faster (elapsed time) than was possible under the initial release of VSORT.

VSORT handles fixed-record-length relative, sequential, and ROSS/V files with support for both span and nospan records. Version 3 now also sorts huge records, which can be as large as 65,024 bytes for user-managed records. For RMS-managed records, the only limitation on record size is that imposed by RMS.

For further information please contact Evans Griffiths and Hart, Inc., 55 Waltham Street, Lexington, MA 02173, (617) 861-0670.

2500th XENIX-BASED SYSTEM SHIPPED BY ALTOS COMPUTER

Altos Computer Systems today announced shipment of its 2500th 16-bit 586 microcomputer running the XENIX™* operating system.

Introduced in October, 1982, the Altos 586 series is a complete family of business systems based on a powerful 16-bit networking microcomputer. The five-user (upgradeable to eight) computer offers networking capability, allowing expansion to over 200 workstations and peripherals.

The system is available in two configurations: the 586-10, with a 12 MB hard disk; and the 586-14, with a 42 MB hard disk. All systems include a single 5½" floppy drive. All can support an additional 42 MB hard disk drive to provide a total on-line capacity of 84 MB. The 586 family is based on the Intel 8086 microprocessor operating at a high

performance 10 MHz, with main memory of 512K bytes, expandable to one megabyte. More information is available from Altos Computer Systems, 2641 Orchard Parkway, San Jose, CA 95134, (408) 946-6700.

*XENIX is Microsoft's microcomputer implementation of the UNIX operating system. UNIX is a trademark of Bell Laboratories.

FILE ARCHIVAL SYSTEM NOW AVAILABLE FOR VAX

The ARCH file archival system provides for the easy off-line storage and retrieval of arbitrary files on a DEC VAX system. ARCH can handle any of the three types of Files-11 formats: indexed, sequential, and relative.

With ARCH, the user can add comments to explain actions and maintain file history. Several features aid the user in searching the archived file to eliminate unnecessary de-archiving to locate a specific file. An internal help facility provides assistance in all phases of the ARCH system.

The ARCH system was developed in 1983 at NASA's Jet Propulsion Laboratory. It is now available from COSMIC, NASA's software dissemination center. Price for source code is \$975.00; supporting documentation is available separately for \$12.50. An abstract describing ARCH capabilities in more detail is available on request. Contact: COSMIC, 112 Barrow Hall, University of Georgia, Athens, GA 30602. Phone: (404) 542-3265.

NAT'L INFORMATION SYSTEMS RELEASES ACCENT R V8

National Information Systems, Inc. announces the release of Version 8 of ACCENT R. Significant new capabilities of this 4th generation language and relational DBMS are the simultaneous concurrent update option, SCREEN option, and new retrieval extensions.

NIS also announces an opportunity for easy and convenient DEC-10/20 to VAX conversion. The VAX version of ACCENT R,

scheduled for delivery in mid-1984, will be syntactically identical to the current DEC-10/20 version. Any site that purchases ACCENT R for the DEC-10 or -20 now will be offered the chance to convert the license to the upcoming VAX for only \$150. For details contact National Information Systems, Inc., 20370 Town Center Lane, Cupertino, CA 95014.

MASS-11 SYSTEM FROM MEC HAS INTEGRATED SPELLING CHECKER

An integrated, interactive spelling checker and corrector that actually checks spelling while you edit is now a feature of the MASS-11-4A word processing package from Microsystems Engineering Corporation (MEC), Hoffman Estates, Illinois.

The most advanced system on the market today, the MASS-11 spelling checker offers DEC system users the ability to check spelling while editing. It features a 50,000 word general dictionary; four special dictionaries of 2,000 words each for math, physics, geography, and computers plus a personal dictionary with unlimited capacity so that you can add as many words as required for your particular needs. For more information contact MEC, 2400 Hassell Rd., Ste. 400, Hoffman Estates, IL 60195, or phone (312) 882-0111, Telex 703-688.

NEW VAX SORT ENHANCEMENTS FROM TOUCH TECHNOLOGIES

Touch Technologies, Inc. announces a major enhancement to TTI SORT, a new high performance sort package for Digital Equipment Corporation VAX/VMS computers.

The package is now implemented using a transparent (unseen) interface between user applications and the VAX-11 SORT. This eliminates the need for any code changes, and allows users to benefit from the speed advantages of TTI SORT immediately.

Installation of the package takes only 5 minutes, and is implemented by re-linking the user programs using a supplied command file. Other software which uses the VAX-11 SORT, such as DEC's DATATRIEVE, can be improved significantly by linking it with TTI SORT.

A perpetual license for TTI SORT sells for \$1,295 for the first CPU, and \$300 for each additional CPU. For further information, contact Touch Technologies at 609 S. Escondido Blvd., Suite 101, Escondido, CA 92025, (619) 743-0494.

CIS ANNOUNCES VMSACTSYS ENHANCEMENTS

Computer Information Systems of Massachusetts, Inc., has announced several enhancements to VMSACTSYS, the company's Resource Accounting and Chargeback System for Digital Equipment Corporation VAX/VMS systems.

VMSACTSYS is a menu-driven, project-level accounting system, enabling users to monitor usage of DEC VAX systems and gather information about resource utlization, such as connect time, amount of disk use, amount of I/O activity, CPU time, and related items.

The price of VMSACTSYS is \$2,995 and is available from CIS at 165 Bay State Drive, Braintree, Mass. 02184. Computer Information Systems of Mass., Inc., is a seven-year old, software and computer services firm.

CJ/ACCOUNTS RECEIVABLE ENHANCEMENTS AID USER

Users of Collier-Jackson's CJ/ACCOUNTS RECEIVABLE can now track a single customer through multi-divisional levels automatically under a common customer number. The new release (No. 1.05.A) has also added multiple user-defined fields and a free-form comment field for more flexible and

customized reporting. This feature allows users to define their own headings, field sizes and edit data for these new fields within the standard system.

CJ/ACCOUNTS RECEIVABLE is a data base oriented, completely online, menu driven accounts receivable system providing control of customer account status for sales and collection analysis. It is designed to run on Hewlett-Packard's HP 3000 and Digital Equipment Corporation's VAX series of computer hardware. For more information about CJ/ACCOUNTS RECEIVABLE contact Collier-Jackson's corporate offices at 5406 Hoover Blvd., Tampa, FL 33614. (813) 885-6621.

DELPHI INFORMATION CENTER INTRODUCED FOR USE WITH VAX

General Videotex Corporation today introduced the Delphi Information Center for organizations with Digital Equipment Corporation VAX computers.

The Delphi Information Center makes the full range of software and services that constitute Delphi, the nation's third largest public information and communications utility, available for private host network operation.

Applications for the Delphi Information Center include communication, information retrieval and transactions, according to General Videotex president Wes Kussmaul.

"The Delphi Information Center builds on the capabilities of VAX computers and VMS operating system software," Kussmaul said. "Users can send electronic mail not only to other users on the private host network, for example, but also to anyone subscribing to The Source or CompuServe. They can also scan the unread messages in their mailbox."

In addition to electronic mail, users linked to the Delphi Information Center can send Telex, TWX, ECOM and Western Union

Mailgram messages directly from their personal computers or desktop terminals.

The Delphi Information Center is priced at \$65,000 for a one-time license and installation support which includes the establishment of all necessary gateways. The cost of hardware necessary to access independent services is not included. General Videotex Corporation is located at Three Blackstone Street, Cambridge, Mass. 02139, telephone toll free 800-544-4005 (in Massachusetts 617-491-3393).

Delphi and Textrieve are trade marks of General Videotex Corporation.

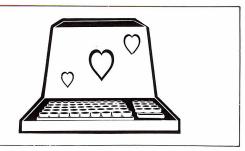
NEW SCOPE VAX RELEASE

Interactive Systems, Inc. announced a new release of its forms management system, SCOPE, for Digital Equipment Corp.'s VAX-11 super-minicomputers. Gary M. Kuba, vice president Sales & Marketing, said "This new version is a result of a survey of over 200 SCOPE customers worldwide and reflects our sensitivity and commitment to our customer's needs."

The new release incorporates an improved user interface, more flexible and friendly programmer calls, and new functionality for the SCOPE Forms Editor. The product will now handle multiple forms per screen and allow for the underlining or boxing of field(s) to enhance the appearance and user's understanding of the screen. Intrafield editing has been added so that a user may change selected characters in a field without altering the remainder of the data. Also included is the ability to create a table of values for the validation of entered data.

These features and others of the VAX version will be included in the new DEC-10/20 release which is scheduled for the first quarter of 1984. For further information contact: Interactive Systems, 131 Middlesex Tpke., Burlington, MA 01803.

PEOPLE PLACES THINGS



DEC SIGNS AGREEMENT WITH SCIENTIFIC CALCULATIONS

Digital Equipment Corporation and Scientific Calculations, Inc., have announced an agreement whereby the two companies will cooperatively market
MicroElectronics Design System
(MEDS™) software on Digital's VAX™
32-bit superminicomputers. The
MEDS software is a self-contained program for automated layout of integrated circuits (ICs) over a wide range of technologies.

Developed for designers of very-large-scale-integration (VLSI) circuits, MEDS automates the layout of gate-arrays, cell-based IC designs, and fully customized VSLI. MEDS placement and routing programs are fully reentrant and dynamic design rule checks are also supported. Productivity improvements of as much as a factor of ten over manual design techniques can be realized with MEDS.

Scientific Calculations, headquartered in Fishers, New York, is recognized worldwide for its SCICARDS™ system for automated design of printed circuit boards and thick-film hybrid circuits.

Digital Equipment Corporation, headquartered in Maynard, Massachusetts, is the world's second largest manufacturer of computers and associated peripheral equipment and related software and supplies.

LANTOR ESTABLISHES NEW VAX SOFTWARE COMPANY

One of Digital's largest West

Coast system houses has formed a new subsidiary to meet the ever increasing demand to economically license VAX software. LANTOR INFORMATION SYSTEMS has begun by supplying users of DEC computers with an affordable means to license the Operating System, Data/Forms Management Software and the Language Compiling Programs. The firm trades exclusively in the VAX/VMS environment. Many companies find the expensive "A" license to be more than they need. They often merely require the right to use the software and obtain distributions. A "DZ" license provides such companies with savings of about 50% over the primary license fee.

LANTOR INC., the parent company, has been in the business of marketing communications systems for the past five years. Further information is available from Matthew Owen, Lantor Information Systems, 8055 Manchester Ave., Playa Del Rey, CA 90291, (213) 821-0642.

EEC SYSTEMS ANNOUNCES AGREEMENT WITH DEC

Digital Equipment Corporation and EEC Systems Inc. announce that they have signed a manufacturing and distribution agreement for LEX-11 word processing and office automation software for DEC's Professional personal computers. A spokesperson for EEC Systems Inc., Marketing Vice President, Mercia Dickman, said that the first product shipment from DEC is expected to be around the turn of 1983-84. However, she added that LEX-11 is

available directly from EEC Systems and that EEC has already been supplying the product to a number of major retail outlets for several months. The LEX-11 software, which is also available for the DEC Rainbow, PDP-11 and VAX product lines is available for \$850 on the DEC Professional under the P/OS operating system from EEC Systems Inc., 327/E Boston Post Road, Sudbury, MA 01776; phone 617-443-5106.

CONTINENTAL RESOURCES NAMED DISTRIBUTOR FOR SPECTRA LOGIC PRODUCTS

The Subsystems Group of Continental Resources, Inc. has been named a non-exclusive distributor for Spectra Logic Corporation single and multifunction controller products. This announcement was made from the office of Robert Hahn, Vice-President of the company's Peripherals Division.

Under the terms of this agreement, Continental is offering the complete line of Spectra Logic controller products. These controllers are designed to plug directly into the mainframe of Digital Equipment LSI and PDP, Data General Nova and Eclipse, Perkin-Elmer, and Texas Instrument TI Line busses. This enables the user to integrate standard SMD compatibility into an existing system. Further information regarding the Spectra Logic controller line can be obtained by contacting Subsystems, Continental Resources, Inc., 175 Middlesex Turnpike, Bedford, MA 01730, 617/275-0850.

LASSIF

Send Classified Ads to: VAX/RSTS Classified, P.O. Box 361, Ft. Washington, PA 19034-0361. \$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

SYSTEMS • CPU's • PERIPHERALS • TERMINALS OPTIONS • MEMORY • COMPATIBLES

BUYING OR SELLING CALL (305) 771-7600

adtaware INCORPORATE TELEPHONE (305) 771-7600

1500 Northwest 62nd Street Suite 512 Fort Lauderdale, FL 33309

Dealers in computer equipment since 1974.



PROFESSIONAL

AVAILABLE ON

MICROFICHE

MICRO PHOTO DIVISION



Everything you always wanted to know about RSTS but were afraid to ask. The RSTS Internals Manual tells all.

REPRINTS REPRINTS REPRINTS REPRINTS REPRINTS!

All content in this publication is copyrighted.

All reprints must be purchased from M Systems, Inc. No other reprints are authorized

All reprints shall contain both a cover and a subscription blank.

Price quotation available on request.

DEC NEW & USED

BUY — SELL — EXCHANGE

Systems • Processors • Memory Options • Peripherals • Modules

LAKEWOOD COMPUTER CORP. (408) 266-2545

COMPUTER REPAIR

PDP11 - All Models

VAX - 11750 & 11780 Terminals - All Models

Micro Computer Systems -

XEROX, IBM, APPLE, KAYPRO, **OTHERS**

Offices In L.A. and SAN DIEGO

Quality work at reasonable rates contract or per call

CALL TODAY FOR FREE QUOTE

ASSOCIATED COMPUTER ENGINEERS A SUBSIDIARY OF COMPUTER SERVICES G

1250 Union Street, San Diego, CA 92101 (619) 233-0103

Your Source for DEC CPU's

VAX 11/750 PDP 11/44 CDC, EMULEX, KENNEDY, CIPHER

(714) 951-7769

P.O. BOX 2000-21 MISSION VIEJO, CA 92690

Digitaldata Inc.

AUTHOR! AUTHOR!

The VAX/RSTS PROFESSIONAL wants you to be an author!

The VAX/RSTS PROFESSIONAL is your magazine. You can make it better by contributing articles, programs or comments directly to us. Our authors are paid honoraria for published works, which because of their hard work, they deserve. We ask you to contribute, send us your manuscripts for possible publication (we prefer machine readable tapes or floppies in PIP, RNO, WORD-11, or ?? format) to VAX/RSTS PROFESSIONAL, P.O. Box 361, Ft. Washington, PA 19034-0361, Attn: Editors. Thank you.

FIXED ASSETS SYSTEM FOR VAX

Calculating depreciation got you down? Know where all your assets are at? Use PLYCOM's Fixed Assets System for software that is easy to use, yet effective. Gives you a complete solution. Includes all the forms, procedures and programs necessary to give management full control over the asset reporting function. Includes excellent documentation and complete support. Features:

- · Easy to use menus
- Book and tax depreciation
- Multi-division or multi-company
- Disposal reporting
- · Property tax reporting
- Depreciation forecasting
- Acquisition reporting Interfaces to General Ledger
- For VAX/VMS

Plycom • services, inc.

P.O. Box 380465 San Antonio, TX 78280 (512) 734-4366

ATTENTION 11/750 11/780

VAX*USERS We have one of the

largest stocks of VAX* Modules Cables

Backplanes **Power Supplies**

in the USA at substantially discounted prices

*Registered trademark Digital Equipment Corp.

INDUSTRIAL SEMICONDUCTORS

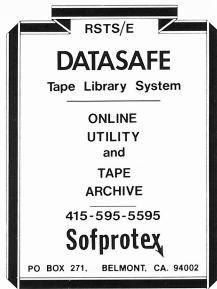
587 Bay Rd, Hamilton, MA 01936 Call us - we probably have what you want

617-468-7177



PLAN PLUS LTD PO BOX 327 LONDON W4 4QD
Telephone: 01-834 7334 × 8328 Telex: 8813271 GECOMS FOR PLAN PLUS
PLAN PLUS U.S.A. PO BOX 662 NORTH TONAWANDA NY 14120 U.S.A.

TANDEM



RSTS RESCUE SQUAD

We salvage all kinds of disasters:

- unreadable disks
- ruined UFDs and MFDs repaired
- immediate response
- telephone DIAL-UP
- on-site
- software tools
- custom recovery
- 90% success to date
- more than 2 GB rescued to date

Brought to you by

On Track Systems, Inc. and a well known (and read)

RSTS expert.

CALL 24 HOURS 215-542-7008

VAX/VMS ON-SITE COURSES

- TIMESHARE USER
- FORTRAN PROGRAMMER

Call toll-free for detailed course outlines. The cost is \$3700 per week.

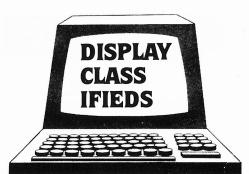
Walter Gallant Mini & Micro Educational Services (800) 824-7888 ext. M4573

VAX AND PDP DEVELOPMENT TIME **Omnicomputer.**™ BUDGET RSTS/E **BYTES** PER HOUR CONNECT TIME 944-9230 Omnicomputer, Inc. 1430 Broadway, New York, N.Y. 10018

BUY, SELL, TRADE: DEC Systems, Parts, Peripherals. Call Paul, Digital Computer Exchange, Inc., 27892 Adobe Court, Hayward, CA 94542. (415) 886-8088.

RSTS/E Application/System Programmers Available. Full/Part time, all languages. RISTIS Optimizers, Box 5942, San Francisco, Cal. 94101-5942

CONSULTING, Los Angeles Area. 7 years experience. RSTS, Basic, Macro, BP2, RMS, Sysgens, some hardware including ENABLE/34. Allan Woloshin, 16032 Knapp St., Sepulveda, CA 91343. (213) 780-2760.



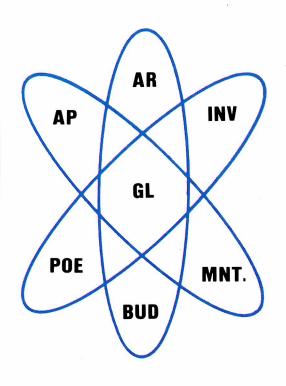
Classified ads are priced at \$1.00 per word. Display ads are \$45.00 per column inch. If we set, this includes border and 2 lines in bolder and/or larger type size, if desired, - please specify.

LIST **ADVERTISERS**

ABLE Computer	
BACmac47	
BACmac	
Computer Power Products 79	
Computer Information Systems 29	
Contel	
Contel31 Data Processing Design, Inc B.Cover	
Dataram Corp 11	
Dataram Corp	
DISC	
Emulex Corp	
Enterprise Technology	
Evans, Griffiths & Hart, Inc 34,49	
Fasbe Group, Inc I.B. Cover	
Finar Systems Ltd	
Froble Enterprises, Dave63	
General Electric	
ILEX Associates, Inc	
Intecolor Corp 61	
Interactive Systems, Inc 51,55	
Joiner Associates63	
Logicraft, Inc 53	
M Systems, Inc	
Manus Services Corp	
McHugh, Freeman & Associates 27	
Micro Processor Associates, Inc 78	
Nationwide Data Dialog 85	
North County Computer	
Services I.F.Cover	
Northwest Digital	
Software, Inc 48-49	
Nyplan 41	
On Track Systems, Inc 60,69,83	
Persoft, Inc	
Personal & Professional 33	
PRO IV	
Ross Systems	
Ross Systems	
RSTS Professional 87	
RSTS Professional 87 Sierra Systems Consulting	
RSTS Professional	

VAX COBOL FINANCIAL SOFTWARE

IBMS™ — FINANCIAL APPLICATION SYSTEMS FOR COMMERCIAL, GOVERNMENTAL AND EDUCATIONAL USERS.

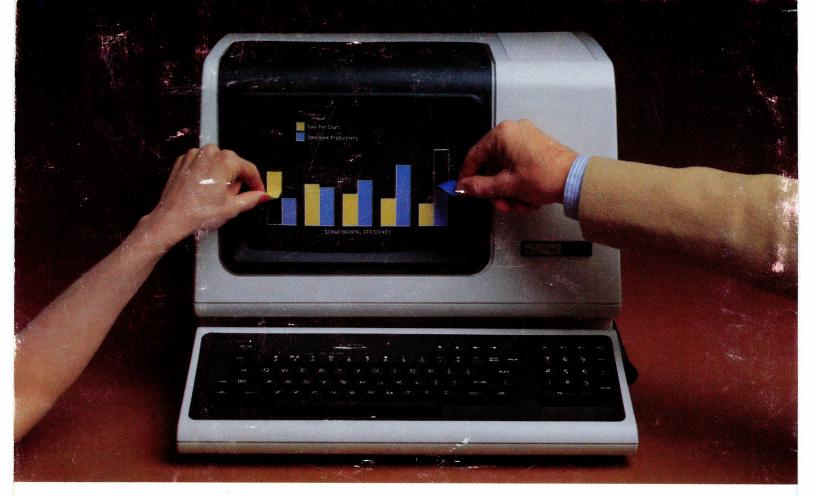


IBMS is the only COBOL based, financial system specifically designed for the VAX. Both the commercial and fund accounting versions of IBMS were designed with a layered architecture for greater efficiency, system integrity and application to networking environments. Of course, IBMS is interactive, integrated and user friendly.



92 Main St, Nashua, NH 03060 (603) 883-3212

TMVAX is a registered trademark of Digital Equipment Corporation.



IB Graph for DEC users. The most cost-effective way to get in touch with your data.

Meet the latest Used Software package from Data Processing Design, Inc. It's called IB Graph.™ Nothing less than the most complete, most cost-effective system for business graphics on Digital PDP-11™ and VAX™ processors.

Of course, you could buy more expensive graphics software, but you'd probably end up with more capabilities than you'd actually need—which wouldn't be cost-effective. Or you could buy *less* expensive graphics software, that simply can't do the job. Again, not very cost-effective. IB Graph is the ideal compromise, giving you literally everything you need, and no more.

IB Graph is a multi-user, interactive businss graphics system that lets you generate graphs in minutes instead of days. And get quality, full-color bar charts, line charts, or pie charts for immediate publication or presentation. With IB Graph, you can better visualize and analyze your data. You can make quicker decisions and increase your efficiency-not to mention your company's profits. Best of all, IB Graph is easy to learn and simple to use, whether you have no experience in computers at all, or are an experienced programmer.

IB Graph outputs to a variety of graphic CRTs and plotting

devices. And it will continue to be compatible with future hardware announcements by DEC.

If you'd like more information, call or write to us at DPD. We'll be happy to tell you more about IB Graph.

IB Graph. You can buy a more expensive system. Or a less effective one.

You can't buy a better one.

