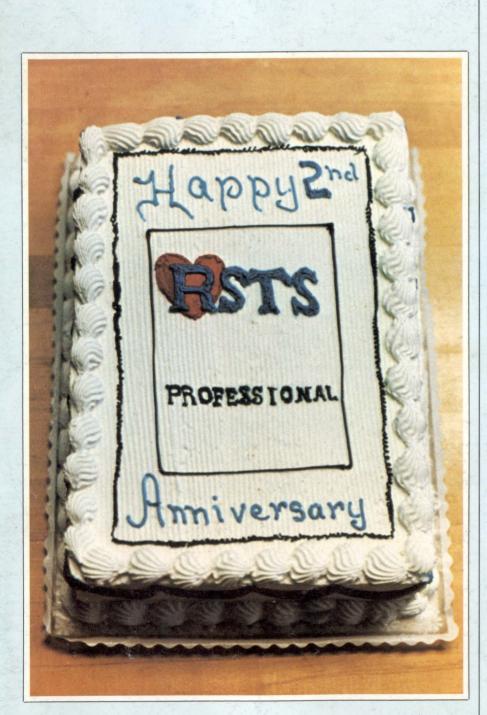
RSTS PROFESSIONAL

Volume 3, Number 4

Second Anniversary Issue

December 1981 \$1000/issue, \$2500/year



INSIDE:

- □ A Top-Down Look at Interactive Software, Part 1
- DEC Data Security, The Software Encryption Solution
- TTYSET Optional Patch for VT100 Width Changes
- □ Software Protection: What Is It Really?
- □ ATPKED: At-pee-kay Basic-Plus Line Editor

TIMER.BAS

- □ BLINK: A Basic Plus Preprocessor
- Sorting Alphabetically
- □ Tips & Techniques
- □ RSTS Defector
- Don't Bubble Quick Sort
- □ The VAX-SCENE

Transportable Software on RSTS/E and VAX/VMS Using a Support Library Technique

- □ Building a Born-Again 40
- Using Sort-11 from Basic Plus-2 on RSTS/E Version 7.0
- Optimizing Space Allocation of Literals in Basic-Plus-2
- **VTEDIT.TEC**
- □ The RSTS/E System Manager
- Dungeon Map
- CALLER.BAS
- System Management for RSTS/E
- □ Benchmark DIBOL vs. BASIC + 2
- Over The I/O Page
- □ RSTS/E Monitor Internals, Part 3
- □ The RSTS Extension
- □ TSXplus An Alternative to RSTS
- □ More . . .

Financial Decision Support Canned For Easy Use!

Fast Flexible Information

MAPS[™] IS FRESH! It's the ideal decision support software product for today's financial manager. With MAPS you get fast, fresh, flexible information handling the moment you want it. Plans, forecasts, models, reports can all be created and manipulated right in your own department. Changes that affect your company can be reflected and evaluated on the spot. And best of all, MAPS is canned for easy use! Call Ross Systems right now. Ask for a demonstration of MAPS. You'll be amazed what a little fresh decision support can do. Currently available on worldwide timesharing, or purchase for DEC's,* RSTS/E and VMS operating systems. Contact Ross Systems for more information today. *DEC is a trademark of Digital Equipment Corporation.

1800 Embarcadero Road Palo Alto, CA 94303 (415) 856-1100

Regional offices: New York. Dallas, San Francisco, Los Angeles

"Now that's fresh decision support."

MAPS meets your daily requirements:

SO FRESH SO FRIENDLY SO EASY

Financial planning 100% Consolidations 100%
Consolidations 100%
Financial modeling 100%
Performance forecasts 100%
Full color graphics 100%

UNAGENENT AND FOR PLANNING STRATEGY

ross

Looking for a quiet, tough printer system? Southern Systems has it. The QT Family.

People have been talking about The QT for a long time. They didn't call it The QT, of course. They just talked about the ideal line printer system — one that would be both durable and easy to operate, a lowcost system with great versatility and dependability. Give us a line printer system, they said, with one answer to my most important needs:

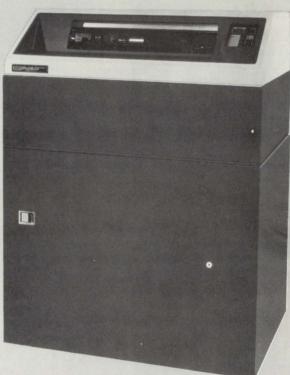
YES

 Do the QT printer systems give me a choice of 300, 600, 1000 & 1200 lpm?
 Does the QT family set the standard

- for quietness? Are they built for production DP
- applications? Are the bands heavy duty steel?
- Operator changeable?
- Are they easy to maintain?
- Is the print quality top-caliber?
- ✓ Is there a diagnostic display? Internal self test?
- Towel ribbon? Forms alignment? Membrane touch control?
- Does SSI guarantee compatibility with my computer?
- Is service available nationwide?
- ✓ Is the QT line printer family affordable and cost-effective?

There's just one other answer you might need: "Delivery in approximately 30 days."

	600-900 lpm1000-pl	us lpm nchronous or asynchronous
My computer system	is	
Company		A REAL PROPERTY AND
Address		
City	State	Zip





(800) 327-5602 • Telex 522135

page 1

CIRCLE 2 ON READER CARD

"we write everything with Ambase"



Don Chick and Greg Jones, Automation, Inc., Omaha, Nebraska, Service Bureau, using the power of AMBASE in an integrated environment with AMCOR's Accounting and Business Control Application products.

"AMBASE training was well presented and made our use of AMBASE easier.

We are using AMBASE to write custom programs and customize modules within other products. It takes less time and we write everything exclusively with AMBASE.

We are able to put out a report in a matter of hours as opposed to days. We couldn't live without it." AMBASE is a revolutionary state-of-the-art system for application development and data base management. AMBASE is increasing programmer productivity worldwide from 100-900%. In addition to the data management system, AMBASE includes a

N	 21	1
$I \setminus I$		T(

Mail to: AMCOR Computer Corp. 1900 Plantside Drive Louisville, Kentucky 40299 Please send more information on AMBASE DEC RSTS Software to: Name

Zip

company	
Address	

Company

City State

Phone

report generator, query language, screen formatter and automatic code generator.

If you would like to find out more about AMBASE, just clip and mail the coupon, or give us a call, TOLL FREE at 1-800-626-6268. We will send you free information immediately.





Contents **2nd Anniversary Issue**

A TOP-DOWN LOOK AT INTERACTIVE SOFTWARE, Part 1	8
"Professor" Al explains how we should build software and why we don't! DEC DATA SECURITY - THE SOFTWARE ENCRYPTION SOLUTION	12
Michael B. Schwartz A good look at how data can be protected.	
TTYSET OPTIONAL PATCH FOR VT100 WIDTH CHANGES	19
Our Meta-Editor tells us more about TECO and VT100's. SOFTWARE PROTECTION: WHAT IS IT REALLY?	22
A continuing section on this timely subject of concern to us all.	
ATPKED: At-pee-kay BASIC-PLUS LINE EDITOR	28
Why didn't DEC think of these? TIMER.BAS	38
Michael H. Koplitz	
Look for Mike all of next year as he tells us some of what he knows. BLINK: A BASIC PLUS PREPROCESSOR	40
Steve Holden	40
From abroad, a valuable implementation.	
\$REORDR — SORTING ALPHABETICALLY	50
I knew REORDR was good for something!	
TIPS & TECHNIQUES Steven L. Edwards	52
A regular column from the expert at Software Techniques.	
RSTS DEFECTOR	54
Joel Schwartz, M.D. No room for his picture this issue. Maybe next time, Eleanor.	
DON'T BUBBLE - QUICK SORT	58
Neil Robertson	
So simple, and so useful! THE VAX-SCENE	
Transportable Software on RSTS/E and VAX/VMS Using a Support Library Technique	. 60
Dave Froble A RSTS-VAX conversion, with lots to tell us, and some new ideas, too.	
BUILDING A BORN-AGAIN 40	65
Dave Mallery	
USING SORT-11 FROM BASIC PLUS-2 ON RSTS/E VERSION 7.0	66
Ron D. Troy We knew it could be done! Here's how.	
Ron D. Troy We knew it could be done! Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed.	68
Ron D. Troy We knew it could be done! Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC	68
Ron D. Troy We knew it could be done! Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed.	68 70
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow	68 70
Ron D. Troy We knew it could be done! Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager.	68 70 72
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky	68 70 72 74
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookir	68 70 72 74
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky	68 70 72 74
Ron D. Troy We knew it could be done! Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNINC: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookir But the rest of you enjoy! CALLER.BAS Steven Fahey	68 70 72 74
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookir But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers?	68 70 72 74 ng. 76
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookir But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller	68 70 72 74 ng. 76
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookir But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller Gary has been managing a system for some time, and has some great ideas	68 70 72 74 79 76 78
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookir But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller	68 70 72 74 79 76 78
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookin But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller Gary has been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers.	68 70 72 74 ^{19.} 76 78 82
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookir But the rest of you enjoy! CALLER.BAS Steven Fahey Wan to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller Gary M. Miller Gary Mabeen managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers. OVER THE I/O PAGE	68 70 72 74 ^{19.} 76 78 82
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookin But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller Gary has been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers.	68 70 72 74 ^{19.} 76 78 82
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookin But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANACEMENT FOR RSTS/E Gary has been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers. OVER THE I/O PAGE Bob "MACRO MAN" Meyer Macro Man shows us another use of resident libraries. RSTS/E MONITOR INTERNALS, PART 4	68 70 72 74 76 78 82 84
Ron D. Troy We knew it could be done! Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookin But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Cary W. Miller Gary has been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcaif Is Dibol faster than BP2? Benchmarks is a popular issue among our readers. OVER THE I/O PAGE Bob "MACRO MAN" Meyer Macro Man shows us another use of resident libraries. RSTS/E MONITOR INTERNALS, PART 4 Mike Mayfield	68 70 72 74 76 78 82 84
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid looking But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary M. Miller Gary bas been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers. OVER THE I/O PAGE Bo' 'MACRO MAN'' Meyer Macro Man shows us another use of resident libraries. RSTS/E MONITOR INTERNALS, PART 4 Mike Mayfield Last in the clasic ''internals'' articles we leam about send/receive. THE RSTS EXTENSION	68 70 72 74 78 82 84 88
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookin But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller Gary has been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers. OVER THE I/O PAGE Bob "MACRO MAN" Meyer Marco Man shows us another use of resident libraries. RSTS/E MONITOR INTERNALS, PART 4 Mike Mayfield Last in the classic "internals" articles we learn about send/receive. THE RSTS EXTENSION Kevin Paul Herbert	68 70 72 74 78 82 84 88
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookin But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller Gary Mab been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers. OVER THE I/O PAGE Bob "MACRO MAN" Meyer Marco Man shows us another use of resident libraries. RSTS/E MONITOR INTERNALS, PART 4 Mike Mayfield Last in the classic "internals" articles we learn about send/receive. THE RSTS EXTENSION	68 70 72 74 78 82 84 88 93
Ron D. Troy We knew it could be done! Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookin But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with cd's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary M. Miller Gary bas been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers. OVER THE I/O PAGE Bob "MACRO MAN" Meyer Matro Man shows us another use of resident libraries. RSTS/E MONITOR INTERNALS, PART 4 Mike Mayfield Last in the classic "internals" articles we learn about send/receive. THE RSTS EXTENSION	68 70 72 74 78 82 84 88 93
Ron D. Troy We knew it could be donel Here's how. OPTIMIZING SPACE ALLOCATION OF LITERALS IN BASIC-PLUS-2 Brad Smith Understanding strings can be crucial to space and speed. VTEDIT.TEC David Spencer THE RSTS/E SYSTEM MANAGER Jeffrey R. Harrow More for the system manager from our regular system manager. DUNGEON MAP Brian Lomasky WARNING: This map is intended for recreational purposes only. Serious Dungeon players should avoid lookin But the rest of you enjoy! CALLER.BAS Steven Fahey Want to do away with ccl's and save small buffers? SYSTEM MANAGEMENT FOR RSTS/E Gary W. Miller Gary Mab been managing a system for some time, and has some great ideas BENCHMARK DIBOL VS. BASIC+2 Frank Metcalf Is Dibol faster than BP2? Benchmarks is a popular issue among our readers. OVER THE I/O PAGE Bob "MACRO MAN" Meyer Marco Man shows us another use of resident libraries. RSTS/E MONITOR INTERNALS, PART 4 Mike Mayfield Last in the classic "internals" articles we learn about send/receive. THE RSTS EXTENSION	68 70 72 74 78 82 84 88 93

Coming . . .

SELECT. Fast Extract Utility

- DIGICALC User Report
- U.K. Commercial SIG Meeting
- Los Angeles DECUS Report
- More on Reborn 11/40's and 11/70's
- Telephone Costs Rival Data Processing Costs?
- "Input Loop" **Programming Technique**
- QUERY.TEC -A Search & Substitute **TECO MACRO with Query Facility**
- FILMAP.BAS
- Link-Time Initialization of BP2 Maps
- Getting the Most Out of Your DEC Field Service

More . . .

From the Editors							. 4
Letters to the RSTS Pr	0						. 6
Dear RSTS Man							36
Tips and Techniques							52
Classifieds							
News Releases							
List of Advertisers							

The RSTS Professional Magazine, December 1, 1981. Vol. 3, No. 4. Published bi-monthly. Single copy price \$10.00, \$35.00 per year. Published by M Systems, Inc., 753 Johns Lane, Ambler, Pa. 19002-0245, telephone (215) 542-7008. Send all correspondence and change of address to: Box 361, Ft. Washington, Pa. 19034-0361. Copyright © 1981 by M Systems, Inc. No part of this publication may be reproduced in any form without written permission from the publisher.

RSTSPROFESSIONAL

From the editors...

RSTS Mr. Howie Brown Chairman IRUS Box 3043 Pawtucket, RI 02861 November 2, 1981 Dear Howie CONGRATULATIONSI With lots of hard work, dedication, late nights and plenty of inspiration you had a great ACCESS-11 conference. The RSTS community and all DEC users salute you and your group for their efforts. We have hoped for some time that there could be a forum for suppliers of services for DEC users to display their products at a dedicated show. INFO, NCC. COMDEX and other shows have their places but somehow the DEC suppliers seem to get lost in the vastness of the expositions. The ACCESS-11 show was big enough to keep people busy, but not too busy to allow lots of one-on-one talking, demonstrations and personalized information. We were pleased to participate in the seminar part of ACCESS-11 for the second year in a row, and from the extra time we spent after the formal sessions talking about the topics we know that they attendees were rewarded for coming to Hartford. Just a word about Hartford, Co.; The Civic center was large, friendly, well placed, responsive (our booth had the right telephone for our modem, and electricity to run it and the terminal) and next to the Sheraton hotel. The hotel managed to feed us when required, gave us a good room to sleep in and had the reservations correct; hard to beat. The RSTS PROFESSIONAL promises to participate in the next ACCESS-11 and hopes that all the vendors and participants concur. Best wishes and thank you Carl B. Marbach Editor, RSTS PROFESSIONAL I have printed a letter we sent to Howie

Brown and his IRUS (Independent RSTS User Society) about their conference and exposition held at the end of October. I hope all of us spend a moment to think about what has transpired. A RSTS show! RSTS people came, RT-11 people came ('we are thinking of moving to RSTS'), RSX people came ('what's a RSTS'), TSX people came, VAX people came and I even met some non-DEC users who came to see what DEC was all about.

I am not sure what the attendance actually was, but we think it was good for all the RSTS vendors. We saw more RSTS users than we will see in one place (except for DECUS) and we had full a easy ACCESS to them, and they had ACCESS to our products and services. The non-RSTS vendors, such as RACAL-VADIC (modems) or BASF (magnetic media) saw fewer people than they ever had in a show before. Did they see enough? They will decide by seeing how many sales leads were generated and what kind they were (lookers? buvers?). As a community we want these vendors back, and we want more: here is what we can do:

> SUPPORT THE VENDORS WHO **COME TO YOUR SHOW!** SUPPORT THE VENDORS WHO **ADVERTISE IN YOUR MAGAZINE!**

Your support means: Ask for information from them (sign up at the show or use the reader service card in the magazine). Give them a chance to show you their products and why they think you should buy them, then make your own decision - we want you to buy what is right for you. If these vendors generate enough qualified action from our show and magazine you will benefit. For example, there were three Statistical (TDM's too) multiplexer companies there. It was a great opportunity to find out what each one has to offer.

The Vendors will be here if we show them we want them; only you can do that. There were so many good things at ACCESS-11, but if you contribute there will be still more at the next.

See you there!

FIRING UP OLD #71 A Case Study

Dave Mallery The 11/70 started volume production in the summer of 1975. My first machine, #104, was delivered in December '75 and has been running like the engine it is, ever since. We recently heard that a local insurance company was selling off its old 70 and by a variety of luck and instinct, go it in the low \$50's. As delivered, it had 1/2 MB of core, 4 DH11 AE's, a TU16 and RWS04(!) and a pair of RPO4's plus a RSTS license and other gems. It had been powered down in March of 1981 and had been sitting in a clean computer room ever since. We sold off the RPO4's and controller for a tidy sum and proceeded to install the rest (less the swapper) in our computer room. My goal was to have the machine shipshape (?) when DEC came in to take it on service. After all, there are 3 or 4 different independant depot repair services today -surely they can fix anything.

Once I had the electrical outlets in place, I re-cabled the TU16's controller and powered her up. Wonderful-she can load address!! Quick-17765000 load address, 60-start-she BOOTS! EUPHORIA. Load a memory diagnostic - EMJA ?? - after a few minutes, reality-parity errors-hard.

Now, fixing a computer without a spares kit requires some creativity and courage, fortunately old memory boxes have many repititions of the same boards just move-em around and you can move the location of the error. At this point, we also discovered a bad regulator in one of the boxes. I used Reliance Electric in Columbus for fixing my regulator and bad memory boards-they seemed to turn them around almost immediately. Now that the memory seemed O.K., on to the major processor diagnostic EQKC ??. This is the one that

... continued on page 36



Editors R.D. Mallery Carl B. Marbach Assistant Editor/Advertising Helen Marbach Controller Peg Leiby Administrative Assistant Hope Makransky Subscription Fulfillment Kathi B. Campione United Kingdom Representative Pauline Noakes RTZ Computer Services, Ltd. P.O. Box 19, 1 Redcliff Street Bristol, BS997JS Phone: Bristol 24181 Contributors William L. Baker Al Cini Steven L. Edwards Steven Fahey **Dave Froble** Jeffrey R. Harrow Kevin Paul Herbert Steve Holden Ed Judge Michael H. Koplitz George T.A. May Mike Mayfield Frank Metcalf Bob Meyer Gary W. Miller Neil Robertson **Robin Robinson** Joel Schwartz Michael B. Schwartz Brad Smith David Spencer Ron D. Troy Cartoons **Douglas Benoit** Photographic Consultant Bill Marbach **Design & Production**

Grossman Graphics

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 publication.)

*This publication is not promoted, not authorized, and is not in any way affiliated with Digital Equipment Corporation. Material presented in this publication in no way reflects specifications or policies of Digital Equipment Corporation. All materials presented are believed accurate, but we cannot assume responsibility for their accuracy or application.

We've got the VT100 market seeing red,



d vellow.

compatible Colorscan 10 for just **and blue**, with performance. Like smooth \$3,195. It's the first affordable \$3,195. It's the first affordable display terminal with the dramatic high-resolution color that businesses need.

The new Colorscan 10 features eight independently selected foreground/background colors (red, green, blue, yellow, cyan, magenta, black and white) crisply displayed in both 80- and 132-column formats. You also get a 128-character set, plus 128 special graphic symbols.

The Colorscan boasts state-of-the-art ergonomics too. You get a tiltable screen, high resolution display, and detachable keyboard with separate numeric pad.

or jump scrolling and split screen/regional scroll. Double-high/wide and double-wide characters. Key-

board selected set-up parameters, self testing, even a CRT saver. All standard. If your business needs more effective financial and sales graphs, charts, histograms, or drawings in 80- or 132-column formats, ask about the Colorscan 10.

It covers the whole spectrum of business applications.

Write or call the smarter terminal maker: (609) 665-5400. Datamedia Corp., 7401 Central Highway, Pennsauken, NJ 08109.

	Colorscan 10	Name		1
	Make me see colors. Please: □ Send complete	Company		DATAMEDIA
	product information.	Address		
	sentative call. Return to: Datamedia	City		
	Corporation, 7401 Central Highway, Penn- sauken, N.J. 08109.	State	Zip	From the smarter
a registered k of Digital	(609)665-5400. RS1281	Phone		terminal maker.

VT100 is

CIRCLE 83 ON READER CARD

RSTSPROFESSIONALRSTSPROFESSIONA

LETTERS to the RSTS Pro..

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

According to DEC's Central Commercial Engineering Group RSTS/E will have certain hardware items phased out over the next two years. Some of these items are of no surprise and little consequence. However, a few of these changes may hurt.

Support of the following will be removed with Version 7.1: KL11, LC11, DC11, LS11, LV11, PR11, and 7-Track Mag-Tape.

A suggested replacement for some of these are (respectively): DL11, DL11, DZ11, LP11, LP11, PC11, and 9-Track Mag-Tape.

The following list of devices will be supported on systems upgrading to 7.1 *only* (no *new* system support) until November 1983: RF11, RS03, RS04, RPR02, PR03, RK05* and RK05F (as system disks), LT33, VT05, VT50, IBM 2741, DP11, and Card Readers.

Some suggested replacements are: RK07/ RM02/3 for medium large disks; RL02 for small disks; LA34/38/120 for the hard copy terminals; VT100 for the Video Terminals; DUP11-replaces DP11.

The Card Reader has no replacement - only option is to move to another form of I/O.

Hopefully with the RSTS/E community informed well in advance this set of changes will not severely impact many RSTS/E sites.

We didn't get your name, but THANK YOU wherever you are.

QUESTION: If Mike Mayfield is offering an Internals Manual for Version 7.0 and 7.1 is coming out soon, will there be some sort of Appendix distributed with the new changes in it? Otherwise a V7.0 manual would be a waste of money. Please send a reply so I can make a decision whether or not to order the book.

Stephen Munyan P.S. My first year and two free issues were fabulous! Keep them coming! My article will be coming soon.

The manual will be in a 3-ring binder for easy updates.

Another call from the wild! Has anybody out there...? Well, this time it's about ... actually, I'll start with some compliments. A great magazine, I only hope that such a forum and exposure for RSTS will prod DEC into a ceaseless output of updates (maybe they'll even improve their - our -BASIC+ compiler, although I am grateful for the time left me by a couple of long TKB's - I learnt TECO!!).

OK, now seriously, does anyone out there know anything about a decent semitic semantic processor? Preferably, AC/DC - both English and Hebrew.

Naturally, we're familiar with the main problems facing the semitic programmer. Right to left I/O, distinguishing between alpha and numeric input - a perennial hassle when writing Hebrew (and presumably Arabic), either the user or program must reverse one of the two. But how does the program decide what to reverse (maybe according to upper or lower case)? Leaving the decision to the user merely creates a problem rather than a solution - I live under the perhaps quixotic illusion that programmers are supposed to solve problems...

In our data entry routines we've provided a variety of options - incorporating most of the above. A recent addition to our library was a right to left input simulation routine, the cornerstone being ODT sub-mode, pretty demanding on the CPU with all those interrupts!

Repeat chorus n+1 times — has anyone out there? has anyone out there? has anyone out there? ...

Gratefully, Ilan Caron, Israel

READERS !?!

Many thanks for a marvellous publication. It is truly informative.

I have terminated employment at CBC Services Ltd., and have taken up with Noble Lowdnes & Partners Ltd. I take pride in wearing my RSTS PRO T-shirt.

> Yours, Graeme McLoughlin London, England

Graeme, although we want to spread the word about our magazine, we suggest you wear a suit to the office the first few days — then wear your RSTS PRO T-shirt.

We find the magazine most interesting and effective. We hope it will help us improve our RSTS systems.

Best regards, J. Peled, Software & Hardware Supportman Nikuv-computers and data processing (1971) ltd. Israel

I just read your piece in V.3, #3, and felt I should respond. Our Center has been running RSTS/E (versions 6B, C, and 7) on an 11/40, for three years and thought I might add a few "watch out for's" to your list.

The serial number on our CPU is 300, (obviously one of the later machines), and until last year, had had *no* ECO's applied to it. Some other hardware items that contribute to the overall scene are:

System Industries (S.I.) 9400 Controller with two CDC 9766 drives, three RK05's, two TU10's, two DECtapes, one DH11, one DZ11, EIS, FIS and a partridge in a pear tree.

We had very frequent problems with our machine turning its "RUN" light off. (As you probably know, the "RUN" light differs in function from other 11-CPU's. It means only that the CPU has control of the BUS.) When we put the machine on maintenance contract, they replaced the CPU backplane which virtually eliminated this problem. So make sure you have the new backplane that is interconnected with a P.C. board, not just wirewrapped. The only way to recover from this failure is to halt, hit START, enable, and reboot.

We found the only way to maintain good performance is to run the S.I. Controller with a throttle count of one (even two won't make it). You can get along with two or four if you don't make frequent accesses to other peripherals and don't mind "data late" errors when you do. (BACKUP doesn't tolerate very many errors of any type.)

FIS may not be a match for FPU, but it *does* save space in a two word floating point math situation. We had recently an occasion to duplicate our patient scheduling system on another hospital's 11/70, (without FPU) and one program had to be overlayed to get it in the address space; such is the difference an FIS makes.

Watch out for the power distribution above the CPU backplane. Those plugs' pins have a tendency to corrode, causing a resistance to the five volt current flow, eventually causing heat buildup which will melt the plug. I understand this has been fixed in machines with serial numbers over 6000. We appreciate the articles in *RSTS Professional*. Keep 'em coming ... and thanks.

Sincerely, Tony L. Wilson Programmer/Analyst Spain Rehabilitation Center University of Alabama in Birmingham

And we appreciate letters such as yours. Thanks to you also.

Several issues ago, a letter appeared in [this feature] regarding the availability of a c.p.u. speedup kit for the PDP11/70. Your reply, in print, was in reference to our model B-11 speedup kits and that evaluations were being made by PDP11/70 sites of which you were aware. You also indicated that an evaluation was "due" in a subsequent issue.

Having advertised in your magazine on occasions since the above "letter" appeared, I thought it prudent to indicate now that over two dozen B-11 units are operational at this time in addition to a significant number of our other product, the Model C-XX Overtemperature Protection Systems.

I would be happy to discuss either of these products or provide customer references to any interested parties who would like to write or call: Nassau Systems, P.O. Box 19329, Cincinnati, Ohio 45219, (513) 231-1283. I am frequently available also in my Boston office, at (617) 735-9704.

I am most interested in establishing the availability, reliability, and appropriatneess of our two products for PDP11's. Your assistance in this endeavor would be deeply appreciated. I might add that advertising in your magazine has produced a healthy level of responses ... approximately a factor of two more than a similar ad placed in *Computer Decisions* and *Computerworld*.

Thank you.

Yours truly, Bruce P. Almich President, Nassau Systems

Thank you, Bruce!

You have an excellent magazine. You get down on DEC a little too hard sometimes — but we're only human (not computers) and we all have faults (very small faults I might add). Keep up the good work. Your tech articles are interesting, especially with hardware (I'm in Field Service). It's good to have more customers know about hardware. It can sometimes cut down on my service calls.

> Thanks, Dave Huffman Digital Equipment Corportion

Thank you for the vote of confidence, Dave.

We have been getting your magazine since the first issue and have found much of interest. Our company has been designing and programming systems using RSTS for over seven years and so have accumulated much information that might be of interest to others.

Since your magazine has been so helpful to us, we thought that some effort should be made to contribute and I enclose a "Quick" sort function that you may think is worth printing.

Yours faithfully, Neil S. Robertson

Manchester, England

We thank you, Neil, and readers can find your sort function (Don't Bubble - Quick Sort) on page 58 of this issue.

In response to Ed Horst's letter to Richard Marino in V.3, #, I would like to outline our experience with adding cache to an 11/34.

... continued on page 81 (Photo Contest on page 65)

Better than Datatrieve!

- Easier to understand
- Much more powerful
- Better manuals
- Better support

Not Limited to RMS-11!

- Can access most file types and DBMS
- Include your own file routines
- Mix file types in a single report

THE ULTIMATE **REPORT WRITER**

GRS

GENERALIZED REPORTING SYSTEM QUERY LANGUAGE and REPORT WRITER

for PDP-11/RSTS systems



Call or write for complete details:

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

Incredibly Flexible!

- Complete formatting flexibility
- Powerful built-in calculations
- User-specified calculations
- Include your own special routines

Easy to use!

- Usable by non-programmers
- English query-language
- Very user-friendly
- Nonprocedural language

A TOP-DOWN LOOK AT INTERACTIVE SOFTWARE

By Al Cini, Computer Methods Corporation

THE CRAFT OF SOFTWARE DESIGN

In "Segmenting BASIC-PLUS-2 Applications" (RP Vol. 3 No. 1), we reviewed the advantages of modular (many parts) over monolithic (all-one-piece) software systems. Functionally organized modules, i.e. modules which exist to perform one isolated independent function, make the "best" systems when they are interconnected using simple coupling mechanisms (preferably, using argument lists with a minimum number of arguments). To many, the derivation of a system's modules is an arbitrary, artistic process - a personal matter between programmer and computer which is closed to objective study. The scientists (we prefer not to think of ourselves as mere "craftsmen," and yearn for the day they grant Nobel Prizes to coders) among us, however, prefer to think of software design as a convergent rather than divergent intellectual endeavor. A number of competent software engineers independently implementing the same set of user specifications will, we believe, produce very similar rather than entirely different results.

Philosophically, this sort of thinking raises an interesting hypothetical question:

"For a given set of perfectly detailed specifications, how many different software designs are possible?"

Basically, there are two different answers to this question: "infinite" and "a few."

If you think "infinite" is correct let me clarify that by "design" I mean the representation of a system in a functionally diagrammatic form, such as the hierarchy charts we discussed in "Segmenting BP2 Applications" (see figure 1). Selection of a programming language and actual coding are not included.

If you still say "infinite" let me further stipulate that all synonymous symbols in the design process (such as different variable names or graphic techniques that mean the same thing) are not to be considered as significant in this exercise.

If you still believe "infinite" is the correct answer, please note that an infinite space is a very comfortable one in which to solve problems for a living. After all, if the solution you select is wrong who can fault you given the astronomical odds against picking the right one?

If you bravely answered "a few," allow me to carry this one step further and assert that the answer is ideally "one." The design of a software system is an essential re-statement of its specifications; where two or more equally plausible design alternatives are possible, the specifications are in some respect incomplete. Of course, all "real world" specifications are incomplete, so we're back to "a few." Software analysis and design the iterative processes by which we gradually converge on agreement between specifications and final product — are still human activities which elude "cookbook" description (if we could define them precisely, then we could program our machines to do them).

Just the same, a few program design **heuristics** which facilitate the development of software systems have evolved and become accepted over the years. One of these, "top-down software development," pursues program construction by defining the higher levels ("main routine") of a system and proceeding through the design of the lowest levels ("subroutines") by a process called **stepwise refinement**.

TELLING TOP FROM BOTTOM

Simply stated, the "top" of an application is the part a user sees, while the "bottom" communicates directly with the machine. Information understood at the top of an application, such as names, addresses, social security numbers, etc., are very abstract from the computer's point of view. By the same token, cylinders, tracks, sectors, buffers and the like seem pretty foreign to the average user. The purpose of a software system is to convert a user's input symbols into the proper output symbols by transparently translating them to and from forms the computer can deal with.

This translation process occurs in discrete steps at specific levels of the software design sometimes referred to as **levels of data abstraction**. Consider the very simple example of a program which reads a sequential file and prints its contents on a terminal (figure 1). At the highest level of the application, a user requests a printed listing of, say, all the **RSTS Professional** subscribers. The computer, which can't deal with entire logical files and complete output reports (setting aside PIP and Datatrieve for the moment), can however manage individual input and output records. At the next level of the application, we deal with the nextlowest abstraction of all **RSTS Pro** subscribers, i.e., **individual** subscribers.

In a stepwise fashion, we have "decomposed" the userworld data into a smaller form by refining the overall file listing process into individual record handling operations, which are executed iteratively for each record in the subscriber file. Since our computer (really, our programming language/operating system) offers instructions to deal with these decomposed data records (GET and PUT/PRINT), no further refinement is required and our design is complete.

Note that data at each level of abstraction flows from left to right within our design (the solid arrows), while program control (the dotted arrows) is passed into and from each refined functional module as it is invoked ("called").

Had our computer lacked a GET instruction, we would further need to refine our "GET SUBSCRIBER RECORD" function into sub-functions able to retrieve and de-block Iteration, where a bounded part of a program is executed repeatedly as long as a specified logical condition is true. Any continuous stream of similar data items implies an

interative or "loop" program structure to process it. A "break" in such a stream resulting in different treatment of data items with certain distinguishing characteristics implies a selection.

physical disk sectors. Without device drivers, we would need to write code at still lower levels to manage seeks to specific cylinders and tracks. RMS elevates our computer's data handling capabilities by providing instructions to handle logical records, and the operating system provides disk-handling services, so we can translate our user's data into a machinemanageable form and back again without very much programming. Likewise, Datatrieve or a DBMS can operate directly on entire collections of logical records and further reduce or in some cases even eliminate the need for applications programming. A cross-section of RSTS/E and some of its layered products, viewed from this data abstraction perspective, is presented as Figure 2.

DATA FLOW ANALYSIS

In earlier articles we reviewed what just about everyone in data processing has been

exposed to by now: structured programming. Without detailing the discipline, we can recapitulate simply that only three control mechanisms are needed to write structured programs.

- Sequence, where program control flows sequentially from one program statement to the next;
- · Selection, where program control is routed into one or another part of a program based on the evaluation of a logical condition:

RSTS/E users, having trouble controlling usage of your computer system? Is there panic when passwords are changed? Have you ever bothered to change your passwords at all? System security is a serious problem. Announcing PASMAN, the intelligent solution.

PASMAN is a REACT account management substitute. Features include:

- · Performs the common REACT functions account add and delete. Easier to use than REACT, and more "forgiving."
- New features never available before. Mass change of passwords for a single account, group, or the whole system! You supply the new password, or let PASMAN make one up.
- Identify users by a logical grouping (such as ACCOUNTING) and with an account description.
- Reports of what account belong to who. Identify user growth patterns, be able to plan ahead for upgrades.

In short, PASMAN will help you to get back control of your system.

PASMAN is available now for only \$350. It comes complete with a through user guide, which can be ordered separately for \$10. A trial version of PASMAN with manual is available for \$50.

PASMAN can also be ordered with DUMPER, a general purpose file dump utility. It is fast, easy to use, and the output is more useful than other available dump utilities. DUMPER is an excellent programming tool that eliminates one-shot programming to examine disk output data.

The purchase price for DUMPER alone is offered for \$150. Like PASMAN, it also comes with a user manual. PASMAN and DUMPER can be ordered together now for the discount price of \$400.

> INFINITY SOFTWARE CORPORATION 2210 Wilshire Blvd. Suite 801 Santa Monica, California 90403 (213) 820-2702 CIRCLE 87 ON READER CARD

For example, lets modify our earlier example to list only **RSTS Pro** subscribers who own or use PDP-11/70s (figure 3). At the "individual subscriber" level, we need to add a function which tests each subscriber record after it is retrieved and returns a true-or-false flag depending on the contents of some "CPU-OWNED" variable in its MAP. Our main program would use the contents of this flag to decide whether to call the "PRINT SUBSCRIBER RECORD" function, and our single input record will thus end up in one of two places: on the listing or in "the bit-bucket." The flow of subscriber records is thus split by our application into two separate data streams, only one of which is printed.

In any software system, an action of some sort is triggered by each input, the result of which may or may not be physically Even apparent. though some subscriber records will be

printed, each will be dealt with according to the rules established in our program. When we count all printed and ignored records, we get a tally of all RSTS Pro subscribers.

Note that, in these trivial examples, the lowest-level blocks can be coded as individual program statements. It is theoretically true that the ultimate refinement of any design is into individual program statements, but in practice any large hierarchy will stop short of these lowest levels.

December 1981

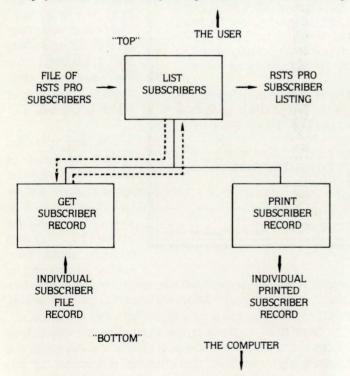
Especially when a project is staffed by more than one person, the "final refinement" will be delegated to the appropriate programmer. Also, data in even very large designs will rarely require refinement to a depth or more than six or seven levels, depending more on the availability of high-level languages and record management software rather than on the nature of the application.

BATCH VS. INTERACTIVE SOFTWARE

Our industry sometimes likes to make arbitrary distinctions between this and that class of software or programmer. From our brief review of data abstraction and program design, we can see that an applications programmer programs for users while a systems programmer programs for other programmers. The engineers who built RMS support payroll software developers in the same sense that they support the Datatrieve development team. We act in a systems role when we build screen formatting or file look-up routines for our fellow programmers, and in an applications role when we use these and similar modules to write budgeting or data entry programs. "Systems" vs. "applications" has nothing intrinsically to do with challenge and job satisfaction; really good programmers can make themselves at home in either capacity.

Likewise, "batch" misleadingly elicits visions of card readers and keypunches while "interactive" suggests CRTs. Peripheral devices are only accidental characteristics of batch/on-line applications: the real distinctions are a matter of software design.

In a classical batch application, someone (the "operator") must gather the required input, schedule the proper computer job, and distribute the output when it is finally produced. The computer job itself is bracketed by



		"ТОР"
logical record collections	{	DATATRIEVE, APPLICATION PROGRAMS
logical records	{	RMS, BASIC-PLUS-2 PROGRAM STATEMENTS
physical disk files	1	the FILE PROCESSOR, PIP utility
physical disk sectors	1	DEVICE DRIVERS. INTERRUPT SERVICE

"BOTTOM"

FIGURE 2. RSTS/E and Layered Products Cross-Section.

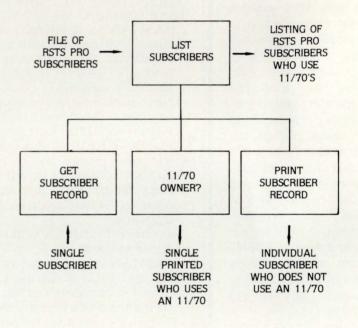


FIGURE 3. List File With Selection.

special control statements ("Job Control Language," or JCL) which initiate the job, assign physical devices to logical units, handle errors, and generally manage "housekeeping" functions.

In an interactive application, the user is his or her own operator. Logon and RUN procedures submitted directly to the machine through CRT commands replace JCL, input is entered as it presents itself and output is often displayed immediately to the user. In the hierarchical design of an interactive application, the "top" levels will solicit and process control information (menu selections), and use these to decide which user functions to invoke.

Many interactive applications have some "batch" characteristics. Imagine our simple RSTS Pro subscriber listing program as an "80/80 card-to-print" utility and the batch nature of this job becomes obvious. But "batch" is a matter of data abstraction rather than of procedures. A software system is neither "batch" nor "interactive" per se. but may support the processing of batched as well as individual data items.

In a future article, we'll examine menu-driven systems from a few new angles.



CALOUT

POOR MAN'S COMPUTER LINK UTILITY

When you run CALOUT from your terminal you may dial up another computer by simply entering the phone number at your keyboard. You are then able to work as a remote user to the other computer with the additional ability to move files between your system and the other computer. All types of files may be transferred between the computers in either direction. This includes binary program files. Complete communication error checking guarantees that the files are transferred correctly even if the telephone lines are noisy. The package may also be used to automate incoming and outgoing TWX systems which then become directly attached to your computer.

CONTRL

REMOTE INTERACTIVE TRAINING AND USER SUPPORT

When you run this utility program at your terminal you are able to capture any keyboard on the system. Thereafter, every keystroke entered by the user, together with every response from their job is also presented to your terminal. In addition, any keystrokes at your terminal will appear at the user's screen and the system will respond exactly as if they were entered by the user. A complete log file of the session may be kept. The process of capture and release is invisible to the user. The user may be engaged in any task whatsoever, indeed the user keyboard may be logged off and turned off when CONTRL captures or releases it.



UTILITY PACKAGE MAKES FOUR TERMINALS OUT OF ONE

With this package you may run up to four independent, interactive jobs of any kind from your terminal with context switching between them at any point in a job session. Each of the four logical terminals generates its own log file of all that passes between you and the computer. This includes straight paranthetical text which you may insert anywhere in the middle of a session without interrupting the job. In the session log file, each key stroke is underlined to distinguish it from computer responses to the job.

- ★ No Installation, Simple to Use
- ★ Clean, Powerful, Certified
- ★ Inexpensive, Low Overhead

Call: Janet (617) 275-6642 Clyde Digital Systems, Box 348, Bedford, MA 01730

DEC DATA SECURITY THE SOFTWARE ENCRYPTION SOLUTION

By Michael B. Schwartz, Prime Factors, Oakland, CA

If you've opted to read this first sentence it most likely means you've keyed on the word "security." The reason: Within the last year the DP community has been bombarded with articles describing known computer crimes, estimates of the much larger number of undetected and undisclosed frauds, and a gloomy outlook at computer privacy in the 80's.

There's good reason for concern. Information is becoming a currency in itself — it has value, sometimes inestimable. Computers are the new concentrators of the information asset. They are the banks of the future. Conventional banks use physical security to separate their currency from those who would like to own it illegally. But the physical security of computers themselves plays a minor roll. A dialup port is a 24 hour a day open invitation and so is absolute access to on and off line magnetic media. Computer crime rarely means physical theft. A simply copy or off-site data draw usually leaves no trace. Similarly, a simple wire tap and data capture into a microcomputer is undetectable. In fact, a number of crimes involved adding information: consider the ramifications of a furtive entry into an accounts payables file to write checks to a dummy company (owned, of course, by the white collar criminal).

The mainframe world has acknowledged these real and potential hazards and is now, rather hastily, trying to patch the breaches in the security wall. The minis are rapidly assuming many of the functions of the mainframe and are encountering the same risks. Risks multiply with the proliferation of distributed processing networks, word processing and electronic mail, minis in use by banks and other financial institutions, and corporations who are replacing their monthly time sharing bill with their own minicomputers.

Yet, until now, there has been no level of data security for the minicomputer user other than the minimal effectiveness of standard operating system passwords and simple access control commands.

This article discusses how a new concept, the GSDEU (Generalized Software Data Encryption Utility) developed by Prime Factors of Oakland, California, and implemented in its PSYPHER[™] system, can solve these problems. This first section is a primer on data encryption. The second will address encryption's roll in security on DEC computers.

As with most techniques of secrecy, the foundations of data encryption grew out of military necessity. A couple of its earliest known forms are the "skytal" device, used by the Spartans against the Athenians; and the Caesar Cipher, used by Julius Caesar against almost everybody. Two world wars brought cryptography out of its adolescence and the emergence of the electronic computer kicked the discipline into maturity.

Types of encryption can be broken down into 2 major categories: ciphers and codes. The most basic cipher, the substitution, works exactly as it sounds. During the encryption process individual letters in an alphabet are consistently substituted for others. The result can appear confusing, but can easily be broken by frequency analysis. Each letter in every alphabetic language has a distinctive frequency of occurance. These frequencies are the joy of the cryptanalyst (one who delights in cracking codes and ciphers. Edgar Allen Poe was notorious for his addiction — read "The Gold Bug"). Take, for example, the following substitution scheme:

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789. becomes 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

Substitution of the word "CLEARTEXT" (means unencrypted information) produces the ciphertext (encrypted information): "2B40HJ4NJ."

It looks cryptic but it isn't. Because of the multiple occurances of 'J' and '4' in the ciphertext one could safely assume that these characters represent the most frequently encountered letters in the English alphabet: 'E' and 'T'. Given a minimal amount of ciphertext even an inferior cryptanalyst can break this cipher.

The second type of encryption, the code, is usually based upon substitution, except groups (not letters) of elements are replaced by an alternative notation. Stenography, where words and syllables are replaced by symbols, is a well known code. Codes based on substitution are as insecure as substitution ciphers and because of their design are particularly unsuitable for computer automated encryption.

Such is the insecurity of substitution techniques. But in one simple step a substitution cipher or code can be made **absolutely** unbreakable. Consider the following harmless series of "random" numbers:

3741226651

Now, using the substitution scheme shown above these numbers can be used to shift the lower sequence of characters to the right just prior to substituting. Thus the cleartext 'C' is no longer '2' but is shifted 3 digits to the right and the ciphertext character becomes '5'. This operation transforms "CLEARTEXT" into "5I8CJPASK." The randomness of the shift has eliminated the tell-tale letter frequencies. Repeated characters in the cleartext are no longer represented by uniRSTSPROFESSIONALRSTSPROFESSIONA

Don't buy a PDP 11"

VAX, PDP & DEC are trade marks of Digital Equipment Corporation

HAMILTON RENTALS 6 Pearl Court, Allendale, N.J. 07401 PLEASE RUSH ME A QUOTATION BASED ON THE FOLLOWING SYSTEM: CIRCLE ONE ONLY IN EACH LINE

 Processor
 Minc
 11/03
 11/23
 11/24
 11/34
 11/44
 VAX 11/750

 Disks
 RX02
 RL01
 RL02
 RK07
 RM02
 RM80

 Operating System
 RT11
 RSX11M
 RSTS/E
 VAX/VMS
 CTS

 Software
 RABBIT
 RSTS
 UTILITIES
 RABBIT VAX
 UTILITIES

	Rental 6 Month	s 12	Months	18 Months
Name				
Position				
Company _				
Address				
City		State	Zij	p
Telephone N	umber			

"...With departmental budget cuts, high interest rates, hardware prices falling back, an upgrade down the road..."

"But I've always bought PDP11s. They're the best."

"I didn't say don't USE a PDP11...just don't BUY one!" "Huh?"

"HAMILTON will rent you the latest in DEC systems – popular 11/23s, hot new 11/24s and 11/44s, and all on a simple no-deposit monthly payment basis. Great for fitting into tight budgets. You'll get prompt delivery, the systems configured the way YOU want them, AND they'll bundle in your choice from HAMILTON's extensive software and utilities library."

"That sounds great, but don't these rental schemes lock you in for 5 years?"

"No, not if you rent from HAMILTON. If you know you're never going to change so much as a terminal for 3-5 years, then go talk to your bank about a long term lease. But, if you want the flexibility of upgrading either all or any part of your system when you need to, then go talk to HAMILTON about rental. They'll listen."

"That's really interesting. I didn't know I had that kind of rental alternative."

"And what's more, you can talk to HAMILTON about their rental-with-purchase-option plan which gives you the right to buy the system when you're ready."

"But will DEC maintain my HAMILTON rental systems?"

"DEC maintenance is already INCLUDED in your monthly rentals. Plus, HAMILTON handles all the shipping and guarantees you a really smooth installation."

"HAMILTON certainly has a comprehensive rental program."

"Of course. They have more experience in renting DEC systems than anyone else, with thousands of installations in Europe and North America."

"You know, if they hadn't limited my department budget this year, I could have gone straight to a VAX 11/750. Hey. I don't suppose HAMILTON rents....."

"11/750s? Of course! Call them now ... Toll Free 800-631-0298 In New Jersey 201-327-1444 In Canada 416-251-1166

or complete the coupon for an individual rental quotation on a system of your choice."



Pearl Court, Allendale, N.J. 07401 One Penn Plaza, New York, N.Y. 10119 415 Horner Ave., Toronto, M8W4W3

LONDON PARIS DUSSELDORF CALGARY

RS

formly identical characters in the ciphertext. This is the famous Vernam or "one-time pad" technique. As long as the series of random numbers is at least as large as the length of the cleartext character stream the cipher is unbreakable. This method is used by the hot-line between Washington and Moscow and it is also believed to be the only method used by Soviet agents. It is called the "one-time" pad because the sequence of random numbers is used only once for each message. While not readily evident, multiple use of the same random number stream will make the cipher increasingly insecure.

Every encryption algorithm must be invertible. That is, once information is encrypted there must be an inverse process to decrypt it. This inverse process is controlled by the crypting (meaning encrypting or decrypting key. The keys for the previously mentioned techniques are straight forward. The key for the simple substitution cipher is the table itself. The stenographer's key is the grammar that describes the ins and outs of stenography. And the key for the one-time pad scheme is the sequence of random numbers coupled with the substitution table.

A key should be what it sounds like: something small and easily hidden and protected. It opens some kind of path to a valuable asset. In this light the stenographic key is useless. It's bulky, and worst of all, publically known. The simple substitution key is not bad, but the algorithm is. The one-time pad's key seems usable and you can't beat the algorithm, but what happens when you want to encrypt a file of a million characters? A key of a million numbers is, for practical reasons, out of the question.

Of the 3 encryption systems presented we see that 2 of them are insecure and one has a key management problem. Furthermore, these examples have dealt with language based alphabetic systems. Just how can encryption be applied to the computer's limited alphabet of 1's and 0's?

The PSYPHER system from Prime Factors solves the key and security problems by employing a variant of the one-time pad process. First, it uses fast algorithms for generating long streams of random numbers. The generation technique uses recursive algorithms. Recursive algorithms are equations that "feed" themselves. Consider the simple equation: $X_{i+1} = X_i + 5$ (this is not a PSYPHER algorithm!). Every value of X depends on the previous value, and the equation must be "seeded" by some initial value, Xo. Let's say that $X_0 = 1$, then $X_1 = 1 + 5 = 6$, and $X_2 = 6 + 5 = 6$ 11 and so on. PSYPHER contains complex banks of specialized recursive formulas that in combination produce a random number stream of a length in excess of 1014 numbers. That is, 10¹⁴ numbers are produced before the sequence begins to repeat itself. This number is called the period of the system.

In operating such a system the user provides the seed (X₀). Each different seed produces a different sequence of random numbers. The series is always larger than any file any user would encrypt. PSYPHER utilizes a double key system, each key an integer between 1 and 2.1 billion. The result is about 4 X 10¹⁸ possible random number streams each greater than 1014 numbers in length. The same "pad" need never be used twice and the stream is always longer than the file. Thus the requirements of the one-time pad are satisfied.

What about the key management problem? This calls for a shift in nomenclature. The one-time pad's random number stream was called the key. Now we will refer to the stream as the "pseudokey" and the seed becomes the key. Thus the requirement of the key is now satisfied: it's compact and easily secured.

Now, how can this ultralong pseudokey be used to encrypt the binary alphabet? The secret lies in one of the most basic computer instructions: the exclusive OR (XOR). This procedure is extremely fast and is invertible. When computer words are XORed against each other a bit for bit comparison is performed. The words are aligned and if the bits in corresponding positions are the same, the XOR result is a '1' in that bit position. If they differ the result is a '0'. To illustrate this let's encrypt and decrypt the cleartext message "A."

0100000101000001 Packed ASCII "AA" XOR 0001101001110010 16 bit random number GIVES 1010010011001100 ciphertext

The ciphertext is clearly no longer ASCII codes and the redundancy of the double 'A' has been masked. To restore the cleartext simply XOR the ciphertext against the same random number.

1010010011001100	ciphertext
XOR	
0001101001110010	16 bit random number
GIVES	
0100000101000001	"AA" again

Now we have a complete encryption system. The PSYPHER system prompts the user to enter 2 keys between 1 and 2.1 billion. This unique key combination initiates the generation of a unique pseudokey which is XORed against every byte in a file until end of file is encountered. To decrypt, the same keys create the same pseudokey which is again XORed against the ciphertext resulting in an error free bit for bit restoration of the cleartext file. Using different keys will initiate a different pseudokey and the result will be a "decrypted" file that is as much nonsense as the encrypted file appears to be. The task is accomplished: the user now holds a simple compact key set that can unlock gigabytes of protected information.

The security of the system depends on the careful mathematical definition of the algorithms that generate the pseudokey. This requires state-of-the-art mathematical design and cryptographic expertise. A simple linear congruential generator, as provided in public program libraries. makes an absolutely insecure random number generation scheme no matter how confused the ciphertext appears. Programs exist that can read ciphertext generated by such a simplistic system and deliver the key in just a few CPU minutes. The PSYPHER system contains a multitude of confusions and diffusions in its random number generating tech-



JOB ACCOUNTING and **PERFORMANCE MONITORING for RSTS/E VERSION 7 USERS**

RABBIT-3 is a complete performance monitoring and job accounting system designed especially for PDP-II RSTS/E Version 7 users. Designed as a stand alone system, RABBIT-3 is written entirely in PDP macro assembler for maximum operating efficiency. Fast and small, RABBIT-3 runs in 5K core with only a 1% (approximate) system degredation depending on the sampling rate.

RABBIT-3 is flexible and easy to use. It's also easy to install. After loading the RABBIT-3 tape or disk, just answer a few questions to tailor the system to your needs. In less than an hour of effort, your RABBIT-3 will be generating complete, detailed user information.

RABBIT-3 is cheap. The basic system is available for \$99/month. Furthermore you buy only the features you need to get your job done. Select from the options listed below:



RSTSPROFESSIONALRSTSPROFESSIONA

niques. Only the most careful analysis has determined that the pseudokey generation processes are virtually untracable and are without period degeneration.

Here I've described only **one** of the algorithms employed in the PSYPHER system. The system offers a spectrum of security algorithms designed to allow the user to select a trade-off between runtime efficiency and desired level of security. Included is the U.S. Government's DES algorithm, an extremely secure cipher that does not employ random number generation techniques. The PSYPHER system is a complete ready to load and run module that performs simultaneous file compression, "in place" encryption of user defined fields within file records and encorporates many other security and convenience features.

Thus far I've discussed some basics of encryption and have described the rudiments of stream ciphering (vs. the government's NBSDES block ciphering technique. The difference is profound on all levels; from theoretical design to program implementation.)

But on to the second part: Just how can the abstract concepts of data encryption be applied to the practical world of automated security? The first step is to more closely examine some reasons for data protection. The following list is a non-comprehensive summary of situations calling for the reduction of data exposure risks.

- Efficient banking practice has evolved an interbank network that transfers \$400 billion daily over 99% insecure communication channels.
- (2) The rapid move towards distributed data processing has necessitated constant data communications.
- (3) The automation of previously manually executed business functions has caused a dramatic increase of access to computer systems by company personnel.
- (4) Electronic mail systems, growing proportionally with word processing systems, are, and will be, in extensive use.
- (5) The information glut has considerably increased the volume of offline and off-site storage of magnetic media.
- (6) Public time sharing installations are widespread and on the increase. This includes companies who, for economic reasons, are opening their computers to outside users.
- (7) In the absence of telecommunication facilities, companies are using the postal and courier services to transport magnetic media.

This list is certainly not complete, but is sufficient to enable a classification of the 3 major areas of data exposure:

- Off-site and offline storage
- Online storage
- Communicated information (telecommunicated or physically transported)

Presently the possible solutions to securing information that fall into these classes are limited to products that aim at shielding only one of the 3 exposure areas.

In mainframe computers online information may be secured by "access control systems," complicated systems

that define who can access what files, when they may be accessed, and from which terminals. This is fine for mainframe installations, but does nothing for the DEC user. Hardware encryption devices have been introduced to protect telecommunications; but that's all they do. And there is no off-the-shelf product for protecting offline and off-site media. The usual solution is physical security.

This leaves the DEC security scenario in a sorry state. Online access control is at best limited and simplistic (password protection) and the cost of the last 2 targeted solutions can be steep if not prohibitive. Encryption communication hardware can cost \$5000 per node and the cost of good physical security, including magnetic locks, secured structures and guard personnel is astronomical.

PSYPHER services the 3 areas of exposure via a **single** product. The guiding principles behind the development of the PSYPHER system were cost effectiveness and generality. Previous to the design of the system Prime Factors determined a set of encryption system characteristics that would meet the needs of generalized security. The result was the aforementioned GSDEU.

The PSYPHER GSDEU is a marriage between specialized encryption algorithms and generalized I/O capabilities. In order to accomplish its full spectrum of security tasks, the GSDEU must be software and have the following characteristics:

- It should run standalone. There should be no need to modify existing software. Installation and full operation is accomplished in hours.
- It should encrypt on a file by file basis servicing virtually all file structures supported by an operating system.
- It should offer the user the option of employing more than a single encryption algorithm. This offers the data manager trade-offs between degree of security and run time efficiency depending on the nature of the data to be protected.
- It should have the ability to partially encrypt file records. Under most circumstances only a fraction of a file's contents (e.g. names and addresses) contain proprietary information. Partial, or "field" encryption saves on run time and provides the distinct advantage of allowing the user to operate (sort, merge etc.) on the unencrypted portions of file records.
- It should offer additional useful and money saving services, such as file compression, that should be performed within any utility that manipulates every byte in a file.
- It should provide more than one kind of encrypted output to facilitate compatibility with communication hardware. In particular, encrypted output should be available in binary stream or character coded forms.
- It should be available for use in bath or interactive modes.

Now, let's examine how the PSYPHER GSDEU can accomplish protection in the 3 areas of data exposure. First, the GSDEU can be used to encrypt all off-line media. On-site physical security may be relaxed because the threat of illegal copy is nullified. The result: reduced costs. In addition, file compression significantly reduces the amount of physical

RSISUsers.

Oregon Software Sort-1-Plus

... I would like to single out two of the many products that I have personally bought and paid for. The first is OMSI Sort Version 1.6. I have used OMSI Sort since its original release as a macro replacement for SQWIK and MQWIK, the old DEC type-1 header sort. It was the best software buy I ever made, yielding staggering reductions in run time. Version 1.6 contains the final additions to a fine product - the macro rendition of XQWIK and OQWIK, the extract and reordering portions of the sort. Amazingly fast and truly flexible. Bravo!

Software **Techniques DISKIT**

The second entrant to my 'Hall of Fame' is a package from Software Techniques called DISKIT.

... Using DISKIT, I created 130 accounts and fully extended their centred UFD's in three minutes and forty seconds (a job that used to take four to eight hours). I then copied the full contents of a 300 Mb RM05 equivalent to this new 'well structured' disk in 45 minutes, optimising clustersize and contiguity in the process. I (for once) was speechless... The documentation is excellent. In short, this package is the 'final solution' to structured disks, eliminating all of the time and complexity and reducing the job to one of the simplicity of a SAVRES.

Editor Dave Mallery reprinted from March 1981 **RSTS** Professional magazine

Recommended by RSTS Professional Now available from RTP.

* Is a complete replacement for \$SORT - no other software required

- * Sorts in excess of 32767 records
- * Handles records across block boundaries

* Up to 50 times faster than

- \$SORT
- *Costs £600-00

- * DSU A disk backup utility for creating optimum structures.
- ***DIR** A complete replacement for **\$DIRECT** that runs twelve times faster

***OPEN** An 'open files' display program ('control F')

- ***LIBRARY** For BASIC PLUS 2 users a library of disk handling routines
- *Costs £1250-00

realtime products 1 Paul Street, London EC2A 4JJ. Telephone 01-588 0667

Call Ed Patient or Steve Collins

OMSI, Sort-1-Plus are trademarks of Oregon Software. Portland, Oregon

DEC, RSTS are trademarks of Digital Equipment Corp., Maynard, Mass.

DISKIT, DSU are trademarks of Software Techniques, Inc. Los Alamitos, Ca

storage space required because of the reduced number of volumes to be stored.

Second, the GSDEU protects on-line files from accidental or purposeful disclosure. It compensates for the inadequate security of standard operating system passwords. A file that is encrypted has its access implicity controlled. That is, the file itself may be accessible, but for all practical purposes its contents aren't.

Third, the GSDEU can virtually replace hardware encryption in telecommunications. In some ways the software solution is better. The absence of hardware means no installation costs, no down time, and the inevitable hardware malfunctions are avoided.

The GSDEU's compression facilities perform functions beyond the scope of hardware devices. Compressed data not only reduces transmission times and costs, but offers greater security. This is due to the fact that repetitive information in encrypted files makes the encoded text more susceptible to being broken.

The GSDEU's encryption algorithms can produce output in binary or character coded forms and records may be fixed or variable length. This allows tailoring the encrypted information to the computer's communications gear rather than the other way around.

This completes the discussion about how the GSDEU can successfully service the 3 major areas of data insecurity; but what about the security of the encryption algorithms and their run time efficiencies? Presently there is considerable debate within academic circles about the degree of security inherent in each of many possible algorithms. This is all debated with little or no concern for the pragmatics of production data processing. Obviously, one does not employ an algorithm that encrypts at 500 bytes per CPU-second to continually encrypt and decrypt a 5 megabyte file. In general, it appears that the relationship between processing time and subjectively estimated degree of security is geometric; processing time increases at a very fast rate with the "more complex" algorithms.

The solution to the security-efficiency problem is to make more than 1 algorithm available to the user. Then, in most cases where information is of extremely sensitive nature and relatively short length, the user employs a slower, more "secure" algorithm. But when it comes time to encrypt a mailing list of 100,000 names, the algorithm with the faster runtime is the likely choice. The ideal GSDEU has a number of user selectable algorithms of different design running from 500 to 10,000 bytes per CPU-second. It is important to allow the user to determine the proper trade-off between runtime efficiency and degree of security.

Let's take a brief look at the application of the PSYPHER GSDEU to the RSTS system. Two of the major data exposure areas, offline storage and data communications are independent of the system and any protection it may provide. Therefore if an installation has weak physical security or is communicating data via any medium. PSYPHER is the only protection against data copy and capture.

RSTS can only offer limited protection to online files via sign-on passwords and rudimentary access control pass codes. Just how effective are these "protections"? Breaches of sign-on passwords are well known and published. High School students have breached both sign-on passwords and file access pass codes on much more elaborate mainframe operating systems. And studies have shown that these illegal penetrations have occured with much greater frequency than have been reported. An estimated 1 in 10 security breaches resulting in breaking federal or state laws go unreported. This is because such occurances would frighten corporate stock holders, drive away time sharing business, and embarass management.

It is this writer's opinion that RSTS security will be breached (if it hasn't been already) when the "work-break cost," the cost of researching and writing the breach code, is sufficiently low compared to the value of the information to be fraudulently added, altered or replaced. Computer crime is a business, the work-break cost is simply the break even point in the cost of perpetrating the fraud versus the expected financial returns. With the increased use of RSTS and VMS systems by large financial institutions, the work-break cost is now very low compared to the illegal gains.

Another problem is the case of the privileged user. Unlike most mainframe installations, minicomputer sites tend to be lax in freely assigning privileged accounts to too many individuals. If proprietary data is present, the risk of someone gaining illegal access grows significantly with the number of privileged users. It's simply too easy for an outsider to learn the proper pass codes for global access, especially when the privileged user often works in common user rooms. The privileged account is an obvious convenience. But it takes only seconds to obtain the proper pass codes with a wire tap or a glance over a shoulder.

Finally there's the problem of the operators console. Any unauthorized individual who gains access to the main console has unlimited browsing power, usually without the need for any pass codes. Highly restricted physical access (other than proper encryption) is the only solution.

In summary, the need for data security is rising. There must be a cost effective tool for maintaining privacy and data integrity. In many cases the law requires (personal, medical and government data) via the Privacy Act that an installation take all available measures to insure data privacy. The PSYPHER system gives the user virtually complete control over his own files. PSYPHER GSDEU encryption insures the user that his proprietary files will remain secure and undisclosed independent of the degree of physical security, the operating system, illegal system penetrations, and misused privileged user access. An investment in encryption is simply paying a one time insurance premium against a theft or accident that one hopes will never occur.

Prime Factors is a software and consulting firm specializing in the area of software data security. Its primary product, PSYPHER, is presently in use on IBM, DG, and DEC computers. Other products available are callable DES modules (DESCRYPTTM), an exact implementation of the National Bureau of Standard's Data Encryption Standard; and PSYSUBS[™], a callable version of the stream ciphering algorithms in PSYPHER.

Michael Schwartz is owner and founder of Prime Factors. His staff consists primarily of mathematical and computer scientists out of the University of California. Interested parties should contact:

> Prime Factors, 6529 Telegraph Ave., Oakland, CA 94609 (415) 654-5090

TTYSET Optional Patch for VT100 Width Changes

By David Spencer, Infinity Software, Inc.

One of the great features of the VT100 is the ability to switch between 80 and 132 column mode. When the VT100 was put on the market, I couldn't wait to get my hands on one. No longer, I thought, would I have to hassle with trying to read wide reports on a VT52.

Well, I finally got a VT100, and things didn't work out like I thought they would. In order to examine my wide listing, I had to hit SET-UP, hit the 132-COLUMN key, hit SET-UP again, and then run TTYSET to change my width to 132 columns. Of course, after about doing this once or twice, it was time to get more sophisticated.

There are several alternatives. One is to always run the VT100s in 132 column mode. Another is to write two little programs as CCLs that do the switch and SYS() call themselves. My solution was to patch the TTYSET program to send an escape sequence to the terminal when you change its width.

The following is a patch to the TTYSET program to switch VT100 terminals automatically to 132 or 80 column mode when the width is changed. This can be done easily because the TTYSET "VT100" macro sets "XON"; a feature no other devices utilize. (Using "XON" to check terminal type is used by DEC software like EDT version 2.)

One new variable has been added to TTYSET. WIDTH% is a flag to prevent sending VT100s the width change sequence each time a parameter is changed. If you wish to leave your VT100 in VT100 mode after switching widths, remove the last escape sequence to switch back to VT52 (< esc >[?2I). One note, this patch has been implemented to send the width change sequence to terminals other than your own if they are specified.

The TTYSET patch can applied with the CPATCH auto-

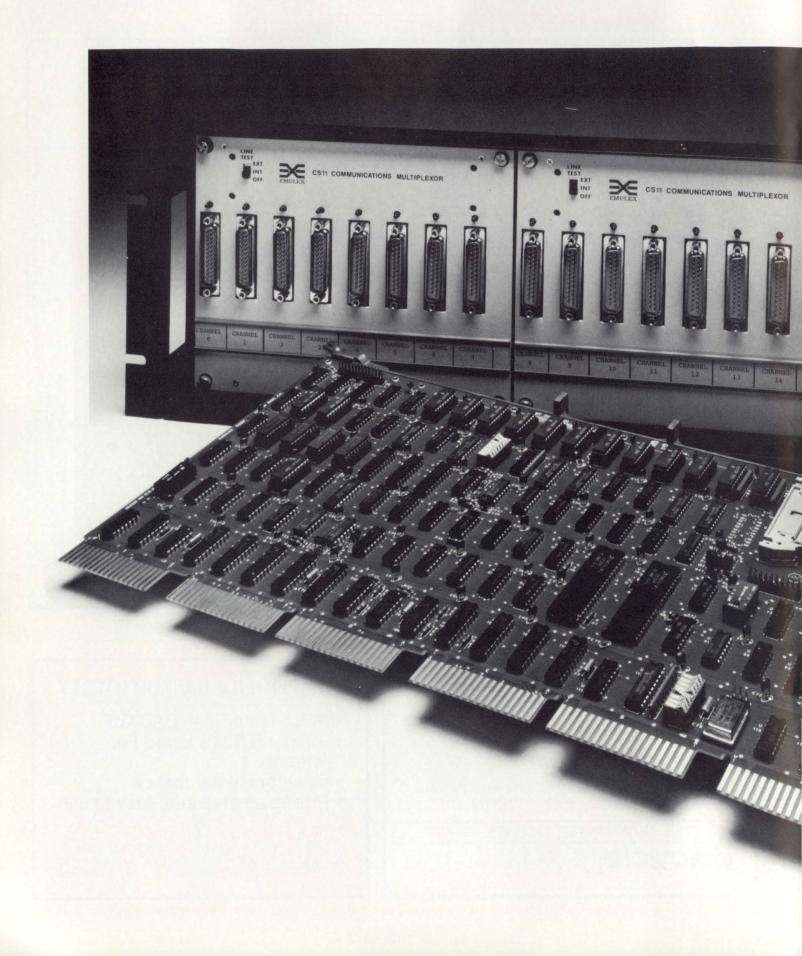
patch program. (If in doubt, see the RSTS/E V07.0 System Manager's Guide.)

*G/2/V <cr></cr>
2! <tab><tab>PROGRAM<tab><tab>: TTYSET.BAS<cr></cr></tab></tab></tab></tab>
*H/1040 <tab>/V<cr></cr></tab>
1040 <tab>GOTO 1530 IF C\$ = "HELP" &<cr></cr></tab>
'l <cr></cr>
WIDTH% = 0% & <cr></cr>
<tab> \ <esc></esc></tab>
•V <cr></cr>
<tab> \ GOTO 1530 IF C\$ = "HELP" &<cr></cr></tab>
*H/1240 <tab>/V<cr></cr></tab>
1240 <tab>GOSUB 12100 &<cr></cr></tab>
*Al <cr></cr>
<tab> \ WIDTH% = V% &<cr></cr></tab>
<esc></esc>
·V <cr></cr>
<tab> \ IF E% = 0% AND V% > 1% AND V% < 256% THEN & < cr ></tab>
*H/1430 <tab>/V<cr></cr></tab>
1430 < tab > GOTO 1500 UNLESS LEN(F\$) & < cr >
*OAI <cr></cr>
1425 < tab > M%(4%) = ASCII(MID(SYS(CHR\$(6%) + CHR\$(9%)).2%,1%))/2% & < cr >
<tab><tab><tab>IF M%(4%) = 255% &<cr></cr></tab></tab></tab>
<tab> \ IF WIDTH% = 81% AND M%(9%) = 255% THEN &<cr></cr></tab>
<tab><tab>C1\$ = SYS(CHR\$(6%) + CHR\$(-5%) + CHR\$(M%(4%)) &<cr></cr></tab></tab>
<tab><tab></tab></tab>
<tab><tab>! GET KB #, GET OUR OWN TO SET UP COLUMN CHANGE FOR VT100 &<cr></cr></tab></tab>
<tab><tab>! IF WE HAVE XON AND WIDTH OF 80 THEN &<cr></cr></tab></tab>
<tab><tab>! BECOME VT100, SWITCH TO 80 COLUMNS, BECOME VT52 AGAIN &<cr></cr></tab></tab>
< <u></u>
1427 < tab > IF WIDTH% = 133% AND M%(9%) = 255% THEN & < cr >
< tab > < tab > C1\$ = SYS(CHR\$(6%) + CHR\$(-5%) + CHR\$(M%(4%)) & < cr >
<tab><tab></tab></tab>
<tab><tab>! IF WE HAVE XON AND WIDTH OF 132 THEN &<cr></cr></tab></tab>
<tab><tab>! BECOME VT100. SWITCH TO 132 COLUMNS, BECOME VT52 AGAIN &<cr></cr></tab></tab>
< <u></u>
<esc></esc>
·V <cr></cr>
1430 < tab > GOTO 1500 UNLESS LEN(F\$) & < cr >
*EX <cr></cr>

David Spencer's VTEDIT.TEC, discussed in the last issue, can be found on page 70.

DEC BEST VALUES	• Much faster than BACKUP • Supports RSTS Large-File					
PRE-OWNED DEC EQUIPMENT						
BUYING AND SELLING	feature					
SYSTEMS • CPU's • PERIPHERALS • TERMINALS OPTIONS • MEMORY • COMPATIBLES	 Ideal for big databases Proven and reliable, easy to use 					
CALL DICK BAKER (305) 979-2844						
Carico Center 2845 NW 62nd Street Ft. Lauderdale, Florida 33309 Telephone (305) 979-2844	CORPORATION Statistics (212) 972-1860					
CIBCLE 49 ON BEADER CARD	CIPCLE 48 ON READER CARD					

Improve your communication



with DEC. Just concentrate.

We've earned our reputation by designing and delivering dependable, easy to live with tape and disk controllers for DEC cpus. Products that really showed the industry how to do it.

Now we've cleaned up the communications I/O morass, with a new DH-11 type multiplexer for PDP-11 and VAX systems. It duplicates all the features of DEC's DH-11. And it goes well beyond, delivering improved line handling capabilities in a far smaller package and at significantly lower cost.

You might say we're now doing for communications controllers what we've already done for tape and disk controllers – applying the latest microprocessor technology and some sexy design to give our customers the performance and reliability they deserve.

Our newest achievement proves the point. To wit: just one hex size board, occupying a single Unibus slot, provides up to 64 lines for terminals, printers or other communications I/O. That's eight times more efficient than the DEC approach.

And our unique system organization means maximum flexibility. You can daisy-chain up to four remote distribution panels, expanding in 8 or 16 line increments all the way to 64 lines. You can configure any 8 line group for an RS-232 interface, with modem control a standard feature. Or you have the option of selecting a current loop interface. And because all your interface variations are made

> The new Emulex CS-11/H really concentrates your communications I/O. Just one hex size board multiplexes up to 64 lines.

at the distribution panels, no additional hardware or cpu restructuring is needed when you add channels.

You get everything you got from DH-11, such as DMA on transmit operation and line format flexibility. Naturally, we're fully software transparent and run existing DEC diagnostics.

But we like to do everything we do just a little bit better than the other guy. So we boosted performance to 19.2 K baud per line. For peak traffic, we can 'double fifo" to 128 characters per 16 lines. Multiple controllers can be added for increased capacity. You get full 16-bit word transfers on DMA operation - not just byte transfers. And, of course, we've taken advantage of our system's high speed bipolar microprocessor to build-in extensive controller and channel self-test and fault indicators

We're concentrating on DEC, making the world's best CPUs even better with some ingenious controllers. We've done it for tape and disk interfacing. Now for communications I/O.

For off-the-shelf delivery (U.S.A.), contact: First Computer Corporation, Corporate Square, 825 N. Cass Ave., Westmont, IL 60559. Telephone: (312) 920-1050.

For complete information (U.S.A.), call or write: Emulex Corp., 2001 East Deere Avenue, Santa Ana, CA 92705. Telephone: (714) 557-7580; TWX: 910-595-2521

In Europe: Emulex Corp., 10th Floor, Cory House, The Ring, Bracknell, Berkshire, ENGLAND. Telephone: 0344-84234; TELEX: 851-849781.



The genuine alternative.

SOFTWARE PROTECTION: WHAT IS IT REALLY?

By Robin Robinson, Mini-Computer Business Applications, Inc.

Mention software protection to most people and they'll say "What?" Mention it to lawyers and they'll say "Right... copyrights, trade secrets, patents, contracts..." Mention it to the guy who spends much of his time actually trying to protect the stuff and he'll groan.

There are relatively few in the software industry who've faced the day-to-day battles straightening out the messes their software got into. A software house is fortunate if they have just one full-time employee chasing down offenders-who run the gamut from not knowing any better to the out-and-out thieves. There's a quiet insanity that accompanies his job, as no sooner does one violation get resolved than another one shows its ugly head. And the truth is, with a large vendor, it's normally not just one at a time. Few understand the dilemma because they don't face it eight hours a day. Hence, it isn't discussed much. But put a few of these "software protectors" in a room together and the chatter will go on for hours. Far more than the safely distant and pedantic recital of "copyrights . . . trade secrets . . . patents, etc." the software protector is put in the trying position of making others understand what these mean sufficiently enough to prevent the need for a courtroom judge to do the educating. Courts and lawyers are, after all, expensive.

THE REAL PROBLEM

The problem faced by vendors and software protectors is that software gets stolen. Surrounding all such thievery is an ignorance of what's supposed to accompany software, and how it's supposed to be protected by those who've been entrusted with it. Users of all types often do not realize that software is usually the proprietary property of a vendor, and that the use of the packages is a licensed use. Frequently, a "seller" represents the transaction as a "sale" and the user proceeds with his use of the software with this understanding in mind. Hence, a consultant or OEM (or even the guy next door) may request to purchase a copy of the users "property" and the user, not knowing differently, will be happy to provide him with one for a small fee. It's not unlikely that the vendor will hear about this years after the fact, and after the software has passed through the hands of three or more companies-with the first two or three of them now out of business. The rhetorical question presents itself to the software protection "Where do you suppose our software is now?"

WHO GETS HURT?

The difficulties encountered are shared by all segments of the industry. Unauthorized copying of computer software

is a contagious disease, compounded by the fact that many of the recipients of the now "illegal" packages don't always know they've been exposed to the potential losses involved. A rather devastating example: Company A purchases a oneuse license from a vendor, and then issues multiple copies of the packages. That's violation number 1. Company A then does a few other things: he sells one of his users a leased machine; he fails to deliver packages he's sold; he fails to ensure the packages are covered by a license or sublicense agreement; he fails to support those packages he does issue; and then he skips town. This example is not too far off the beaten track. What Company A has done is he left a large handful of users holding the bag. The vendor is faced with a group of understandably irate users, many of whom may have source code. Despite an excellent contract designed to prevent this sort of thing, the problem still faces a vendor as to what to do now? Ignoring it may be the easiest choice, but to do so runs the risk of weakening its trade secret protection. Besides, it's not unlikely that the users have formed a sort of modern-day vigilante committee who seek justice where there apparently is none, they've already paid so much money to Company A, they aren't about to pay the vendor for the packages and support they still need, at the vendors' usual rate. And in the wings are probably one or two consultants ready to take advantage of the situation. Try asking any of these users what they think of the software industry now and you're likely to get run out of town. Ask the vendor, and the other consultants involved, while you're at it. Situations like these don't leave any fond memories.

EFFECTS ON END-USERS

On a smaller scale, end-users are frequently left in the cold when it turns out that their supplier (OEM) didn't have the rights he claimed to have when the packages were issued to the users. Often, the user doesn't receive a written agreement for use of the proprietary software and therefore doesn't realize it belongs to some vendor located miles away in some other state or country. It usually isn't until negotiations between the OEM and end-user breakdown that the user discovers he's gotten hold of some "hot" software - information that's often given him by the new consultant he calls in. The vendor gets a phone call, which is frequently how he finds out the original licensee was messing around, and now the software protector has two problems on his hands: an end-user who maybe has his source code and no license agreement (and a bitter taste in his mouth); and an OEM or licensee who's been violating his license agreements. The details always vary, but can range from a user being out anywhere from \$100 to \$50,000 for the software alone. Sometimes more. The software protector will often hear the

AARDVARK SOFTWARE introduces the complete **UCSD PASCAL** system available under RSTS/E for the PDP-11 family.

Not merely a compiler!

This system was designed to facilitate software development and create an ideal Pascal teaching environment. Designed and developed by John Hayward, Ph.D. of Wheaton College, Wheaton, Illinois.

ADVANTAGES

*Based on enhanced UCSD PASCAL, utilizing:

- long integers
- strings
- segmented procedures

*Simplified user interface-automatic linking *Powerful screen editor

*Compatible with UCSD microcomputer systems

*Supports "random access" of files

*Includes Assembler

Price: \$995

Write us, or call toll free for information: 1-800-558-1538

AARDVARK SOFTWARE INC. **783 North Water Street** Milwaukee, Wisconsin 53202

UCSD PASCAL is a trademark of the Regents of the University of California. PDP-11 and RSTS/E are trademarks of the Digital Equipment Corporation.

CIRCLE 41 ON READER CARD

tears in the users voice when he calls in his story. There are hundreds of stories like this which fall in the lap of software protectors. One even hears about a demonstration package (object code only) actually being sold to an unsuspecting end-user with the assurance that this applications package would solve his needs. The seller collects more than \$7,000 from the user and goes on his merry way. The user soon discovers problems with his package and, after failing to reach his original supplier, contacts the vendor. After hearing the first hundred or so tales like these, it's no longer news. Just another story in a long string of abuses.

EFFECTS OF OEMS

The unethical blackmarketeer (as well as those who don't know any better) almost invariably leave behind a computer system that won't function and a user who's quarterly statements are due tommorrow. As software, in general, is mysterious magic to those who don't know how it works, it takes a certain amount of trust for a user to hire a consultant on in the first place. Once he's had an experience with a software thief, or a software buffoon, he's not all that likely to be so easy to get along with when the next OEM or consultant comes along. And, he's probably warning his local colleagues about those "computer guys . . . who'll take you for all they can get." The honest and ethical consultant is not only faced with these effects, but are also faced with trying to keep their business afloat (if not thriving) against the competition only a blackmarketeer can provide (the thief having little overhead as he didn't have to pay for the rights to the software to begin with, and he's not paying royalty fees either). Confused and bearing the brunt of this storm, the ethical OEM may begin to feel it's wiser to follow in the footsteps of his competition.

Not all do, of course, but again the stories abound. One OEM caught another consultant at one of his end-user sites stealing disk packs which contained a vendors software with whom the OEM had signed a license which included a clause requiring the licensee (OEM) protect the trade secret and confidential nature of the software. The OEM later wrote the vendor about the occurance, explaining he'd have stopped the thief if he hadn't noticed the revolver tucked into the thief's belt. A more ordinary method of approach has been for an OEM to contact the vendor asking if one of his competitors is properly licensed for use of the vendor's software. This approach is frequently not used, however, due to the OEM's wish not to become involved in any potential legal proceedings.

EFFECTS ON THE SOFTWARE VENDOR

Not only is a vendor faced with the competition provided him by other software vendors, he can also find himself in competition with his own pirated software. A licensee who's acquired source code from a vendor may modify the packages to such an extent that he feels the packages can no longer be considered the property of the vendor. Hopefully, the vendor has a strong contract to the contrary, but in attempting to enforce it after the fact of distribution by the "ex-licensee", one loses some leverage, to say the least. The one advantage in a situation such as this is the fact that there's a license somewhere that can be discussed in court, if need be. But if a software pirate has gotten copies of the code, and jazzed it up with a few bells and whistles, a vendor is faced with having to acquire a copy of the packages by one means or another just to ascertain whether or not it belongs to him. And if it does, the vendor will likely be spending the next few years in court.

ALIENATED SOFTWARE PROTECTORS

Unless one's software runs hand-in-hand with another vendors packages, those who spend their time trying to protect software (in-house) don't really know who one another are, and consequently, can't help one another out. Hence, there's no easy access to individuals who are trying to protect software for languages other than those used by the vendor for whom the protector works. And those who do protect the same language as one's own, are one's competitors. There's a whole network of computer systems and software packages that contains, from a protection standpoint, relatively few crossroads. The result is a number of individuated cells of activity, each trying to accomplish the same end result in terms of protecting one's own software, and never having the opportunity to really talk about the problem, or at least have a bigger look at the picture. And the picture is a big one.

IS THERE A SOLUTION?

Unbeknownst to one another, there is very likely a small army of software protectors and vendors who've each developed methods which could be useful to the others in the field. There are those who've virtually become experts in their day-to-day software protecting; experience which could easily be shared. Beyond this, however, is an even bigger opportunity: that of combining the common interests of protecting software (no matter who it belongs to) from those who are ignorant about the "rules" and those who could care less about them. Hoping for legislative protection and assistance, or depending on one's lawyer, fails to address the problem realistically, and it's losing software vendors valuable time and money . . . time in which the ignorant and blackmarketeers alike could conceivably cripple the industry. No one individual can really straighten out all the messes their software can get into, at least not in the current environment. True, there are "tricks" which can be utilized to prevent theft, but there are those who are just as fascinated with undoing the tricks as a programmer is in developing them.

The needs of software vendors and protectors have not been fully identified as a whole, but the need for a combined effort has more than adequately presented itself. Computers are a remarkable contribution to the business world. The software that makes those fancy systems play music to

a businessman's ears is a miraculous development requiring hours and hours of sweat and brilliance to produce. The vendor who has cared enough to produce a package (or 50 of them) that will make the lives of others easier is to be commended. True, he's in the business to make a living for himself. Maybe he'd even like to get rich. But if he's any good at all, he's not riding on the wings of someone else's creativity and hardwork. He's producing something new, and hopefully better. And it's costing him a good chunk of money to do so. To then release this into a marketplace full of users who see the value (threatened with thieves who also see the value) places the vendor in an unenviable position. His software protection people (or person) have to be sharp; have to know where to look at what to listen for: and then have to act fast when something's amiss, without having to resort too often to outside attorneys and courtrooms who could very well eat whatever profits come in.

Software protection individuals and efforts could well use some organization. The marketplace (endusers and OEM's alike) are in dire need of understanding the situation and its ramifications. Without adequate protection, both by the courts and on a day-to-day basis, the honest and productive vendor could be forced to close his doors. An Association for Software Protection (ASP) has long been needed, but has only recently been formulated by three software vendors who share the same problems. It is a bright idea needing support and contributions. Theoretically, it could prove to be a vehicle which introduces software protectors to one another and help them to create a marketplace in which a blackmarketeer wouldn't dream of

treading and in which end-users know the rules up front; before the troubles begin. The form and activities of such an association would obviously need to respond to the input received from other software vendors and protectors; information which, due to the present alienation, has previously lacked any organization. Those interested in participating

at the mini-computer level, are invited to contact the author (in writing) at 117 South Brand Boulevard, Glendale, Calif. 91204; or Ms. Adrienne Webb, Digital Information Systems, Inc., 6247 Fair Oaks Boulevard, Carmichael, Calif. 95608; or Mr. Dale Coleman, S&H Computer Systems, Inc., 1027 17th Avenue South, Nashville, Tennessee 37212.

LOOK at the "tear-out" cards in this issue.

There's a "HOT to COLD" form for rating our articles. Let us know what is most and least interesting to you. There's a FREE gift for you. Bring in new subscribers and collect rewards. See "BOUNTY HUNTERS" card. There's a READER INQUIRY card for your convenience.

PDP-11[®] & VAX[®] users...

ComDesign's TC-3 Concentrates Multiple Terminals and Printers on a Single Telephone Line **Cost-Effectively** for the PDP-11 and VAX.

CONCENTRATION Many terminals-one line

- PROVEN BELIABILITY TC-3s are in use world-wide
- STATISTICAL MULTIPLEXING Efficiencies to 400%
- BUILT-IN SYSTEM MONITORS Real-time data and signal displays

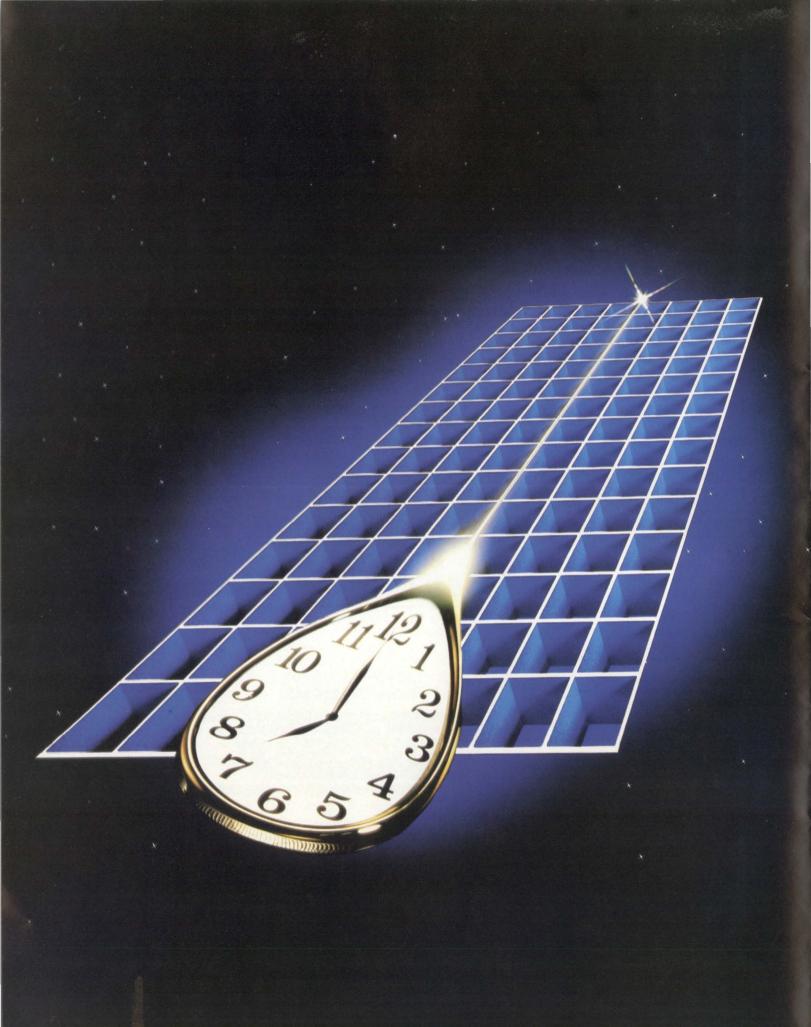
- COST SAVINGS TO 40% Totally eliminates DZIIs and cables
- EBBOB COBRECTION Data checked with all errors eliminated automatically
- FAST ECHO RESPONSE -at last !
- POWEBFUL DIAGNOSTIC PANEL Modem loop-back, unit self-tests, and terminal diagnostics

Put a TC-3 on Your UNIBUS and Get a DZ11 with Stat Muxing too!

"We are impressed . . . have already recommended the unit to a number of other RSTS/E users and shall continue to do so" Intercontinental Forwarders, Inc. See us at DEC USER SHOW, Booth #123

For further information and the sales office for your area Call toll-free (800) 235-6935 or in Calif. (805) 964-9852





Pascal-2® The Time Machine

The First Dimension: Performance

Pascal-2 performs.

Pascal-2 programs run as fast as FORTRAN IV-PLUS programs—or faster. (FORTRAN IV-PLUS is Digital's fastest PDP-11 high-level language.) Pascal-2 code is as small as code generated by any Digital PDP-11 compiler or interpreter—or smaller. And Pascal-2 typically compiles at 1000 lines per minute.

The Second Dimension: Structure and Portability

As a programmer, you can write in a language close to your thoughts. With Pascal's structured methods, you can do the job right the first time. It's easier to *design* in Pascal than it is to *debug* in FORTRAN, assembler, BASIC, or COBOL.

As a software manager, you will see the value of Pascal in improved communication among team members: they can understand one another's code. Pascal's portability will protect your software investment: your programs will outlive your current hardware.

The Third Dimension: Tools, Tools, Tools

The compiler precisely reports typographic or syntactic errors. The interactive, source-level debugger helps detect deep-rooted logical errors. The profiler helps identify code that can be rewritten to speed program execution. Also included are formatters, index generators, and documentation aids— a total of 70,000 lines of Pascal code.

Our 2,000 customers use Pascal for such diverse applications as general ledger and payroll, integrated circuit design graphics, word processing, typesetting, and off-track betting; for trimming integrated circuits, monitoring particle accelerators, real-time ballistics modeling, and controlling saws in a lumber mill.

The Fourth Dimension: Our Past and Future

The core of our technical group has been together more than a decade. Our Pascal-1 compiler entered commercial use in 1975. Before releasing our PDP-11 product, we delivered Pascal-2 under contract to two major computer manufacturers for three different processors. Now we're moving Pascal-2 to Motorola's MC68000, to Digital's VAX-11, and to the UNIX operating system. We're committed to Pascal for the long term.

Call or write. We'll send benchmark details, a product description, and a free copy of the Pascal-2 manual (specify RSX, RT-11, RSTS/E).

PDP, VAX, RSX, RSTS/E, RT-11, and FORTRAN IV-PLUS are trademarks of Digital Equipment Corporation. MC68000 is a trademark of Motorola Inc. UNIX is a trademark of Bell Laboratories.

Pascal-2: The Dimensions of Performance

Ask for a free 18" x 24" poster of this photograph.

Network Computer Services 6 Cunningham Street Sydney 2000, Australia Telephone: 211-2322 TLX: 25523

Valley Software, Inc. 6400 Roberts Street, #390 Burnaby, BC Canada V5G 4G2 Telephone: 604-291-0651 Hourds Computing, Ltd. 7-8 Mill Street Stafford ST16 2AJ, England Telephone: 0785-44221 TLX: 36540

Real Time Products 1 Paul Street London EC2A 4JJ, England Telephone: 01-588-0667

TLX: 884971

d 4316 Shinjuku-ku Tokyo 160, Japan TLX: 24208 AC Copy Kurbrunnenstrass

Kurbrunnenstrasse 30 D-5100 Aachen W. Germany Telephone: 0241-506096 TLX: 832368

Rikei Corporation 1-26-2 Nishi Shinjuku

CIRCLE 60 ON READER CARD

Periphere Computer Systems GmbH Pfalzer-Wald-Strasse 36 D-8000 Munchen 90 W. Germany Telephone: 089/681021 TLX: 523271



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

ATPKED: At-pee-kay BASIC-PLUS LINE EDITOR

By William L. Baker, Pig Improvement Corporation

1.0 ABSTRACT

This article describes the implementation of new commands for RSTS/E version 7.0 utility ATPK. The commands will take some of the unfriendliness out of the otherwise friendly BASIC-PLUS environment by combining the functionality of ATPK with editing of basic without the use of an editor program.

2.0 BACKGROUND

The principal advantage to BASIC-PLUS is programmer productivity through the interactive program development. Productivity is sacrificed when the programmer must constantly switch back and forth between basic and an editor. In the native BASIC-PLUS environment the programmer has the choice of retyping the line or using an editor after saving the program. The inefficiency of the procedure is quite pronounced when learning or modifying large programs with multi-line basic lines. The ATPKED modification provides a simple substitute command to improve programmer productivity.

3.0 MAKE NEW FRIENDS

The ATPKED modification provides the programmer with an extended program development environment including: editing, control-C interruption, logging, run-time system switching, and indirect command files.

3.1 Commands

3.1.1 STANDARD ATPK COMMANDS

The programmer should be familiar with ATPK. See "RSTS/E V7.0 Release Notes," September, 1979 Sequence 17.1.1N. Pay close attention to the section on control characters, quoted control characters, the use of control-M on page 4 of 9, and "Other features" on page 9 of 9.

3.1.2 NEW COMMANDS PROVIDED BY ATPKED

\$EXIT Abort ATKP and close the PK job. This command replaces the use of control-C to end a command file from the KB: and simulates the native environment for control-C. \$X Execute to last \$Substitute command. This command provides the ability to change the same string several times.

\$S line-number Xstring-1Xstring-2Xinteger-1Xinteger-2

Substitutes the integer-1 occurrence on integer-2 line of string-1 with string-2; X is a delimiter and may be any printing character; the space(s) are optional between the "\$S" and the delimiter. The basic line is listed upon successful edit. The "?" is removed from error lines.

The basic line number may be modified by the \$Substitute. This will cause a clone to be generated and listed.

3.2 Examples of Operation.

ATPKED EXAMPL = KB:

Ready

NEW EXAMPL

Ready

10 ! THIS IS SOME TEXT & ! AND SOME MORE

Ready

- \$S 10 /E /E MORE /
- 10 ! THIS IS SOME MORE TEXT & ! AND SOME MORE

Ready

\$X

10 I THIS SIS SOME MORE MORE TEXT & I AND SOME MORE

Ready

\$S10/ MORE///2

10 ! THIS IS SOME MORE MORE TEXT & ! AND SOME &

December 1981

RST_PROFESSIONALRSTSPROFESSIONA

Ready

\$S10/10/100

100 ! THIS IS SOME MORE MORE TEXT & ! AND SOME &

Ready

100	PRINT LOOP \	SLEEP 2 \ GOTO 100
RUN		
EXAMPL	02:26 PM	15-Sep-81
LOOP		
LOOP		
LOOP		
tC		

Ready

4.0 INSTALLATION

4.1 Patching ATPK

Apply the patch to ATPK.BAS source containing any DIGITAL supplied patches. This patch was based upon edit level: V7.0-07B 8-JUL-80.

4.2 ATPKED

Append the ATPKED modification to the patched ATPK.BAS. Compile as you choose considering non-priviledged users etc.

4.3 Considerations

4.3.1 Do not trade an editing bottleneck for a some other bottleneck. That is, you may not have sufficient pseudo keyboards available and you must consider that the total overhead (extra job, memory, etc.) may be greater for ATPKED.

4.3.2 ATPKED is a tool to extend BASIC-PLUS and not a replacement for EDT.

4.4 Helpful Hints

4.4.1 CREATING INDIRECT COMMAND FILES UNDER ATPKED

When creating indirect command files under ATPKED enter two underscore characters before ATPK commands lines to prevent ATPK from executing the commands immediately, example:

PIP TEST.CMD = KB: ____\$@TEST1 ____\$WAIT †Z

4.4.2 ENTERING BASIC-PLUS LINES

When entering BASIC-PLUS lines under ATPKED enter an underscore character before a line beginning with "!" as ATPK will not send comment lines to the pseudo keyboard.

1 EXTEND!		**** ATFKE	D APPEND CODE ****		
		DELETE LIN	E 1 BEFORE APPENDING TO ATPK.BAS.		
50	! WLB	11-SEP-81	*** NEW COMMAND MODIFICATION *** &		
i	410		ON VARIABLE DEFINITIONS &		
1	810 1070-1079	SUB CONSTA	ION SUBROUTINES &		
1	3099-3300 4100-4400	NEW COMMAN CTRL C RES	ND INTERPRETER ROUTINE & START &		
1	13000-13999	Sub ROUTI	NE &		
1	14000-14999 16003	\$S EDIT RO JUMP TO BA	SIC READER AND SUPPRESS CTRL C ECHO &		
1	16030-16500 19030-19032	BASIC LINE TRAP ROUTI			
			8		
55!	*****	******	***************************************		
-	* * W4	TCH FOR CO	MMING ATTRACTIONS * &		
1	* * \$R	RESTORE TH	* & HE BASIC LINE THAT WAS CHANGED * &		
i	*		T \$Substitute COMMAND. * &		
1	* \$L/	OPEN/CLOSE	X & & & & & & & & & & & & & & & & & & &		
1	* \$Fn	FULL SCREE	* & N EDIT. * &		
1	*		* 8		

410!	*** NEW COMP	AND MODIFI	CATION VARIABLES *** & & & & & & & & & & & & & & & & &		
1	L.* SUB.BASIC.LINE.M		CAL VARIABLES DOCUMENTED IN CONTEXT & ASIC LINE NUMBER USED IN \$Substitute &		
1	SUB.CR%	PO	SITION OF THE END OF LINE CHATACTERS &		
1	SUB.CR\$ SUB.ERRMSG\$		NSTANT, CARRIAGE RETURN & ROR MESSAGE SENT TO ERROR ROUTINE &		
!	SUB.HITS% SUB.LF\$	CO	OUNT OF OCCURENCES OF THE SEARCH STRING		
1	SUB.LINE\$	NE	NNSTANT, LINE FEED & W LINE EXTRACTED BY BASIC LINE READER &		
1	SUB.ND.EXTEND%	SW	VITCH TESTED BY BASIC LINE READER: & 0 = EXTEND MODE LINES CONTINUED &		
1			WITH CR-LF		
1	. *		1 = NO EXTEND MODE LINES		
1			CONTINUED WITH LF-CR-NULL &		
1	SUB.NOEX.ERR%		NITCH SET BY BASIC LINE READER TO TELL & HE \$SUD ROUTINE THAT EXTEND LINE WAS &		
1			SJECTED IN NOEXTEND MODE. &		
	SUB.POS%		A STRENT POSITION WITHIN A TEXT LINE		
1	SUB.REF\$ SUB.REF.COUNT%		HE TEXT TO REPLACE THE SEARCH TEXT & HE OCCURANCE OF THE SEARCH STRING &		
1		TH	AT IS REPLACED &		
1	SUB.REP.LINE.NO	SE	HE CONTINUED LINE NUMBER WHERE THE & & & & & & & & & & & & & & & & & & &		
1	SUB.SAVE.COMMANI		AVE AREA FOR THE LAST \$S COMMAND & & AVE AREA FOR TEXT TO REPLACED & &		
1	SUB.SW% SUB.TEXT\$(100)		WITCH TO ACTIVATE THE BASIC LINE READER STRAY FOR THE BASIC LINE TEXT &		
1	SUB. TEXT%	CO	UNT/INDEX OF BASIC TEXT ARRAY		
-	SUB.TEXT.REM\$		RTION OF PK RESPONSE SAVED UNTIL THE & XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
			8		
810!	*** NEW COM	AND MODIFI	CATION SUBROUTINES *** 8		
1	14000-14999		SIT THE \$S COMMAND LINE		
1	\$SUD COMMAND ED	T	8		
1	14950 MESSAGE ROUTINE	SE	END MESSAGES TO THE USER &		
1			8		
1	16500 TEXT SAVER	SA	AVE AND COUNT BASIC LINE TEXT &		
1			8		
1070!	*** NEW COM	AND MODIFI	CATION CONSTANTS *** &		
1075	SUB.CR\$=CHR\$(13) SUB.LF\$=CHR\$(10)	0 1	CARRIAGE RETURN &		
'	SUB.LF\$=CHR\$(10)	() I	LINE FEED &		
2025	GOTO 3100		8		
	I JUMP 1	O THE OUR	COMMAND INTERPRETER &		
30991	******* NEW COM	MAND INTER	PRETER ROUTINE ******* &		
1	THIS ROUTINE WILL TEST THE INPUT CONTROL LINE FOR THE NEW				
1	COMMANDS:		8		
1		\$EXIT	ABORT THE PK JOB AND ATFKED &		
1		\$5 \$X	SUBSTITUTE STRING FOR NEW STRING & EXECUTE THE LAST \$Substitute &		
7100	TE LEET OTO				
3100		.COMMAND\$=	S* THEN & & & & & & & & & & & & & & & & & & &		
1	GOTD 13000 8				
	! TEST FOR \$Substitute COMMAND &				
	I GOTO	HE \$SUB CO	> FOR THE \$Xecute COMMAND & & MMAAND ROUTINE & &		

DEC Terminals,

Peripherals and Supplies?

Veve gotem.

page 30

And we'll get 'em on their way to you right away usually within 24 hours. Exactly what you wanted. At the right price. In time. On time. Every time. Anywhere in the continental U.S.A. You can count on us. Because we're The Suppliers.

DEC Peripherals

BA11KF DD11CK/DK DH11AD DL11E DZ11-A,B,C,D,E,F **FP11-A** MS11LB MS11LD MSV11DD DLV11J RM05-AA RK07EA RK711EA RL02-AK **RL211-AK** RL01-AK RL11-AK RLV21 RLV11 **RX211-BA** RXV21-BA TJE16-EA TWE16-EA TWU45-AA KDF11-AD, HD, HF, HH, HK **DR11C** MR11EA VT103-AA, BA Call us at (904) 434-1022

DEC Terminals

IA 34 **DECwriter** IV LA 36 DECwriter II LA 38 DECwriter IV LA 120 DECwriter III LA 180 DECprinter I VT 50 DECscope CRT VT 52 **DECscope** CRT VT 100 DECscope CRT VT 103 DECscope CRT VT 132 DECscope CRT

Call us at (904) 434-1022

DEC Supplies

Disk Cartridges and Packs Diskettes Diskette Storage Units Mag Tape Ribbons Paper Forms Caddy Anti-Static Rugs Print wheels Terminal/Printer Stands Forms Bursters and Decollators **Check Signers** and a whole lot more

Call us for our complete catalogue: 1-800-874-9748. In Florida, call (904) 434-1022.



We're The Suppliers

114 East Gregory Street Pensacola, FL. 32501

page 32

December 1981

8

RSTSPROFESSIONAL

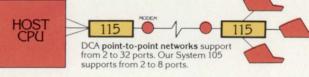
3120	IF LEFT(CTRL.LINE\$,2%)=*\$X* THEN & CTRL.LINE\$=SUB.SAVE.COMMAND\$ &	13480	IF SUB.FOS%=0% THEN 13550 ELSE
•	GOTO 13000 &	`	SUB.HITS%=SUB.HITS%+1% IF SUB.HITS%=SUB.REF.COUNT% THEN 13600
	I TEST FOR \$Xecute COMMAND \$ I RESET CURRENT COMMAND TO LAST SUB COMMAND \$ I JUMP TO \$Sub COMMAND ROUTINE \$	1	ELSE SUB.POS%=SUB.POS%+LEN(SUB.SRCH\$) GOTO 13460
8150	IF LEFT(CTRL.LINE\$,5%)=*SEXIT* THEN 19200 8		I SKIP OUT TO TRY NEXT LINE IF NO MATCH.
	I ABORT ON SEXIT COMMAND		GOTO REPLACE ROUTINE IF DESIRED OCCURANCE FOUND OTHERWISE ADJUST POSITION PAST CURRENT HIT
300	60TD 2030 & &		I GO LOOK FOR THE NEXT OCCURANCE
	I NO MATCH FOUND ON DEC OR OUR COMMANDS SO SEND THE & I LINE TO THE PK JOB	13550	NEXT L.LINE%
1001	**** CONTROL C RESTART ROUTINE **** 8	175/0	! END OF LINE SEARCH LOOP
	THIS ROUTTINE ATTEMPTS TO RESTART ATEK AFTER THE USER HIT S CONTROL C AND AT BEST SIMULATE THE "REAL WORLD" ON THE PK JOB, S	13560	SUB.ERRNSG≸≕'Text exhausted before match' GOSUB 14950 GDTO 2000
200	WE GET HERE THRU THE TRAP AT 19032.		! SET PROTEST THAT SEARCH WAS UNSUCCESSFUL ! GO SEND ERROR ! GO BACK FOR ANOTHER COMMAND
	ERR.FLGZ= 0 \$ EKROP.CHECNZ= 0 \$	13600	SUB.TEXT\$(L.LINE%)=LEFT(SUB.TEXT\$(L.LINE%),SUB.POS%-1%)+ SUB.REF\$+RIGHT(SUB.TEXT\$(L.LINE%),SUB.POS%+LEN(SUB.SRCH
	IN. ATX= 0 & & & & & & & & & & & & & & & & & &		MAKE THE SUBSTITUTION
	LAST.CHR\$=** & & FK.STATUS2=0 & &	13650	IF ASCII(SUB.TEXT\$(1))=63% THEN
	FUT.COUNT%= 0 & & & & & & & & & & & & & & & & & &		SUB.TEXI\$(1%)=RIGHT(SUB.TEXT\$(1%),2%)
	WARNING.CHECK%= 0 & & & & & & & & & & & & & & & & & &		I CHECK IF RASIC LINE WAS AN ERROR LINE I AND REMOVE THE "?"
	! FOUND AT THE START OF ATPK AT LINE 2000 AND & ! ASSUMING THE CONTROL FILE IS "KB!", &	13660	IF L.LINEX<>1% THEN 13700 ELSE L.NUM%=1%
300	SUB.SWX=0X		L.NUMX=L.NUMX+1X WHILE MIN(SUB.TEXT\$(12),L.NUMX,12)==0 AND MID(SUB.TEXT\$(12),L.NUMX,12)<==9 SUB.BASIC.LINE.NOX=VAL(LEFT(SUB.TEXT\$(12),L.NUMX-12))
	SUB.TEXT2=02 3 SUB.NOEX.ERR2=02 3		GOTO 13670 IF SUB-BASIC.LINE.NOX=0% GOTO 13700
	I RESET MODS SWITCHES		! CHECK IF THE FIRST LINE WAS CHANGED
400	L.D\$=SYS(CHR\$(6%)+CHR\$(-4%)+CHR\$(PK.NB%)+CHR\$(3%)) & & & & & & & & & & & & & & & & & &		! FIND THE NUMBER OF DIGITS IN THE BASIC LINE NUMBER ! RESET THE LINE NUMBER JUST IN CASE IT WAS CHANGED ! CHECK FOR NULL NUMBER AND COMPLAIN IF FOUND ! SKIP ERROR ROUTINE
	S I PASS THE CTRL/C ON TO THE PK JOB WITH A FORCE SET NO ECHO SO THAT ONLY DHE ≏C IS ECHOED GO CLEANUP RESPONSES FROM TH PK GO TO MAIN LOOP S S S S S S S S S S S S S	13670 \ \	ON ERROR GOTO 19000 SUB.ERRHSG≸="Modified basic line number is invalid" GOSUB 14950 GOTO 2000
30001	**** \$Substitute COMMAND ROUTINE ****		I RESET ERROR EXIT AFTER BOMB ON 13660
	THIS ROUTINE WILL CONTROL ALL PHASES OF EXECUTING THE \$Sub:		! COMFLAIN TO USER ! RETURN FOR NEXT COMMAND
	1. EDIT THE COMMAND ARGUMENTS. \$ 2. READ THE BASIC LINE INTO AN ARRAY. \$ 3. FIND THE POSITION AND LINE FOR THE SEARCH STRING. \$ 4. MARE THE SUBSTITUTION. \$ 5. REMOVE THE '?' FROM ERROR BASIC LINES \$ 6. CHELN IF THE BASIL LINE PUMBER WAS CHANGED (CLONED). \$ 7. RETYPE THE MASIL TO THE PK. \$ 9. PELIST THE BASIC LINE. \$	13700	GOSUB 2300 FOR L.LINE%=1% TO SUB.TEXT% CTRL.LINE%=SUB.TEXT\$(L.LINE%) GOSUB 10000 NEXT L.LINE% GOSUB 2200 CTRL.LINE%='LISTNH'+NUM\$(SUB.BASIC.LINE.NO%)+CR.LF\$ GOSUB 10000
3100	60SUB 14000 &		! SET NO LOG TO 'RETYPE' BASIC LINE ! START LOOF FOR EACH CONTINUED LINE
	I GO EDIT THE \$S COMMAND		I SET UP NEXT LINE TO SEND I SET UP NEXT LINE
3110	GOTO 2000 IF SUB.BASIC.LINE.NOZ=0% 8		I END LINE LOOP I RESET LOGING
	I ABORT COMMAND IF EDIT FASSES ZERO LINE NUMBER &		I SETUP RELIST OF EDITED LINE I SEND LIST REQUEST AND DISPLAY RESPONSE FOR USER
3200	CTRL.LINE\$=*LISTNH*+NUM\$(SUB.BASIC.LINE.NO%)+CR.LF\$ % SUB.SW%=1% %	13990	6010 2000
	SUB.TEXTX=0% & SUB.TEXT.REM\$=** &		! RETURN TO GET NEXT COMMAND
	60SUE 10000 \$	13999!.	
	I SETUP LIST COMMAND TO PASS TO THE PK I SET SUB SWITCH TO TELL ATKP LOG ROUTINE TO ACTIVATE STORE DEDUCTION FOR A STORE AND	14000	**** \$Sub COMAND EDIT ROUTINE
	! THE BASIC LINE READER & ! ZERO THE TEXT LINE COUNT ! RESET THE TEXT LEFT OVER FROM THE PREVIOUS PK RESPONSE &		FORMAT OF \$Substitute COMMAND:
	9 GO SEND THE LIST COMMAND AND COLLECT THE BASIC LINE &		\$\$(spaces)BLN(spaces) <d>\$\$<d>R\$<d>C<d>L</d></d></d></d>
3210	IF SUB.NOEX.ERR% THEN SUB.NOEX.ERR%=0 \$ SUB.NOEX.ERR%=ND TEXT FOUND IN NOEXIEND MODE* \$		WHERE:
	605UB 14950 8 60T0 2000 8		() INDICATES OPTIONAL NON INMPORTANT SYNTAX BLN BASIC LINE NUMBER DELINITED CHARGETER FOR ALL APPLIMENTS. THE FIRE
	I TEST FOR PROBLEM WITH NOEXTEND MODE AND EXTEND BASIC S I RESET ERROR SWITCH S I SETUP ERROR MESSAGE S I SEND MESSAGE S I GO BACK FOR ANOTHER COMMAND S		delimiter character for all arguments. The first 15 Assumed for all following arguments St search string Rt replace string Courance where replace is to occur (Default=1) Contided line number to start search (Default=1)
3400	SUB.HITS%=0% 8		
	INIT COUNT OF OCCURENCES OF THE SEARCH STRING	14040	SUB.BASIC.LINE.NO%=0% SUB.SRCH4=**
3410	0 FOR L.LINEX=SUB.REP.LINE.NOX TO SUB.TEXTX & SUB.FOSX=1X &		SUB.REF4=** SUB.REF.COUNT%=1%
	I START LOOKING FOR SEARCH STRING STARTING AT THE REQUESTED & CONTINUED LINE NUMBER &		SUB.REP.LINE.NO%=1% ! INIT ARGUMENTS AND SET DEFAULTS
	SET STARTING LINE POSITION TO 1 & &	14050	L.C\$=CVT\$\$(MID(CTRL.LINE\$,3%,LEN(CTRL.LINE\$)-4%),8%)
13460	SUB.FOS%=INSTR(SUB.FOS%,SUB.TEXT\$(L.LINE%),SUB.SRCH\$) &	1	L.FLD%=1% L.FLD%=L.FLD%+1% WHILE MID(L.C\$,L.FLD%,1%)>=*0*

FROM STATISTICAL MULTIPLEXORS NETWORKS, ΔΤΔ

DCA protects your initial investment in statistical multiplexors with the lowest-cost network growth in the industry. So you can start with a small datacomm net-

work today, and expand or modify it to meet your needs tomorrow.

Our System 115 statistical multiplexor can be used in point-topoint networks to support from 2 to 32 asynchronous terminals at a



remote site. DCA's statistical multiplexing assures excellent response time, character transparency and error-free transmission.

DECHOST

DCA's multipoint multiplexing configuration serves a number of remote terminal locations with just a single telephone line, for substantial savings in phone-line and hardware costs. DCA users

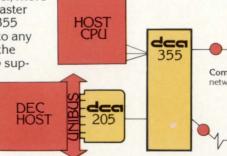
have benefited from multipoint multiplexing since January 1979.

DCA's System 205 is a statistical multiplexor designed for DEC UNIBUS*-based computers. The 205 requires only one UNIBUS slot to emulate up to 16 DEC

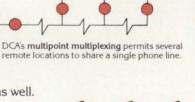
DZ11 modules and a 128-port stat mux. This greatly lowers hardware costs and improves response time as well.

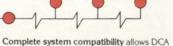
Because of DCA's unique modular design, all of the above networks

could easily expand into larger, more powerful networks. As the master network processor, System 355 gives terminal users access to any host computer anywhere in the network. In addition, the 355 supports up to 126 ports, 44 of which can be highspeed synchronous trunk links. Several 355's can be combined to greatly expand this support. Features include port conten-



205





networks to grow without growing obsolete.



tion, switching, unlimited routing, X.25 support and a network management system built right into the product.

Total system access and unlimited network growth at very low cost - that's the DCA advantage. For complete information, call or write for our 16-page brochure.



*TM Digital Equipment Corporation

16040

DIM SUB. TEXT\$ (100%)

RESERVICE STORAL RESERVICE STORAL

SUB. BASIC.LINE. NOZ=VAL(LEFT(L.C\$,L.FLD%-1%)) I ARRAY TO SAVE THE BASIC INF I STRIP OFF \$5, CR-LF, AND LEADING SPACES I SET FIELD INDEX TO 1 I LOOK FOR END OF BASIC LINE NUMBER I SET UP ERROR MESSAGE I EXTRACT BASIC LINE NUMBER AND CHECK ALL NUMERIC 160491 *** AUTOMATIC EXTEND/NOEXTEND MODE ROUTINE *** THIS ROUTINE FINDS THE FIRST LINE AND DETERMINES IF THE END-OF-LINE SEQUENCE IS A CR-LF (EXTEND MODE) OR LF-CR-NULL (NOEXTEND MODE). GOTO 14900 IF SUB.BASIC.LINE.NOZ=02 14055 GOTO 14900 IF SUB.BASIC.LINE.NOX= L.C\$=CVT\$*(KIGHT(L.C\$+L.FLD%),8%) L.DLM*=LEFT(L.C\$+1%),8%) SUB.ERRMSG*= Delimeter not found* GOTO 14900 IF L.DLM*='* L.C\$=RIGHT(L.C\$+2%) 16050 TO\$=SUB.TEXT.REM\$+TO\$ GDTO 16100 IF SUB.TEXT% L.CR%=INSTR(1%,TO\$,SUB.CR\$) IF L.CR% = 0% THEN SUB.TEXT.REM\$=TO\$ GOTO 16010 I CHECK THAT BASIC LINE NUMBER WAS NOT ZERO I STRIP BASIC LINE NUMBER OFF OF THE COMMAND AND I STRIP OFF LEADING SPACES I TAKE THE NEXT CHARACTER AS THE ARGUMENT DELIMITER I SET UP DELIMITER ERROR MESSAGE I CHECK THAT WE HAVE A DELIMITER I STRIP OFF DELIMITER APPEND ANY TEXT FROM THE PREVIOUS PK RESPONSE TO THE CURRENT RESPONSE SKIP OUT IF NOT IF LINE 1 HAS BEEN FOUND LOOK FOR A CR IF NO CR SAVE THE CURRENT TEXT 14060 L.FLD%=INSTR(1%,L.C\$,L.DLM\$) L:rEUX=1MSIM(14/E.(D*FL.DLM*)) SUB.SRCH#=LEFT(L:C#.L.FLD2-12) SUB.ERMSG4="Search string not found" G010 14900 IF SUB.SRCH#="* AND LEN(SUB.SRCH#)=02 L.C#=RIGHT(L.C#.E.FLD2+12) IF MID(TO\$,L.CR%-1%,1%) = SUB.LF\$ THEN SUB.ND.EXTEND% = 1% 16060 ELSE SUB.NO.EXTEND% = 0% I LOOK FOR THE END OF THE SEARCH STRING I SAVE THE SEARCH STRING SETUP EXROR MESSAGE ABORT IF SEARCH STRING IS NULL STRIP OFF SEARCH STRING FROM COMMAND LINE ! IF THE LF IS BEFORE THE CR ASSUME NO EXTEND MODE ! OTHERWISE ASSUME EXTEND MODE 16100 GOTO 16300 IF SUB.NO.EXTEND% ! TEST FOR NO EXTEND MODE CR-LF-NULL CONTINUED LINES L.FLD%=INSTR(1%,L.C\$,L.DLM\$) 14070 16119! *** EXTEND MODE ROUTINE *** 16120 SUB.CR%=INSTR(1%,TO\$,CR.LF\$) SUB.CRX=0% THEN IF LEN(TO\$)=0% THEN 16010 ELSE LOOK FOR THE END OF THE REPLACE STRING I SOUR FOR THE END OF THE REFLACE STRING I SETUP ERROR MESSAGE I ABORT IF NULL REPLACE STRING AND NOT TRAILING DLM I FAKE POSITION INDEX IF NO DELIMITER I SAVE REFLACE STRING I STRIP OFF REFLACE STRING I LOOK FOR THE END OF A LINE IF THERE IS NO END LINE AND NO TEXT THEN RETURN ELSE SAVE THE TEXT FOR THE NEXT RESPONSE FROM THE PK AND RETURN GOTO 14100 IF L.C\$= .. 14080 L.FLD2=LNSTR(1%,L.C\$,L.DLM\$) L.FLD2=LEN(L.C\$)+1% IF L.FLD%=0% SUB.ERRMSG\$=*Replace count invalid* 16130! NOTE: NOW WE HAVE A COMPLETE LINE FROM THE PK. SUB.LINE #=LEFT(TO\$,SUB.CR%+1%) 16140 I TAKE DEFAULTS ON REST IF COMMAND IS NULL I LOOK FOR THE END OF THE COUNT ARGUMENT I FAKE DML IMDEX IF NOT FOUND I SETUP ERROK MESSAGE TO\$=RIGHT(TO\$,SUB.CR%+2%) IF INSTR(1%,SUB.LINE\$, "Ready") THEN SUB.SW%=0% GOTO 16010 SUB.REP.LOUNT%=VAL(LEFT(L.C\$,L.FLD%-1%)) I EXTRACT THE LINE WITH END SEQUENCE 14085 STRIP LINE FROM FK LINE IF KEADY FROMFT IS FOUND WE HAVE ALL OF THE TEXT AND RESET THE SUB SWITCH AND SUB.REF.COUNTZ=1% IF SUB.REF.COUNTZ=0% L.C\$=RIGHT(L.C\$,L.FLD%+1%) ! GET AND TEST NUMERIC COUNT ! SET DEFAULT=1 IF ZERO ! STRIP COUNT FROM THE COMMAND 16160 GOSUB 16500 GOTO 16120 GOTO 14100 IF LEN(L.C\$)=0% SUB.ERRNSG\$=*Replace line number.invalid* SUB.REF.LINE.NO%=VAL(L.C\$) GOTO 14900 IF SUB.REF.LINE.NO%=0% 14090 I GO SAVE CURRENT LINE IN THE ARRAY I REPEAT UNTIL NO ADDITIONAL LINES ARE FOUND IN PK I SKIP LINE IF NO COMMAND IS LEFT I SET UP ERROR MESSAGE I GET AND CHECK NUMERIC THE LINE NUMBER I ERROR IF LINE NUMBER IS ZERO 163001 **** NO EXTEND MODE SUBROUTINE **** SUB.CRZ=INSTR(1%,TO\$,LF.CR.O\$) 16320 IF SUB.CR2=0% THEN IF LEN(TO\$)=0% THEN 16010 ELSE CONTINUE 14100! 14890 GOTO 14990 · SKIP VAL(x) ERROR ROUTINE *** \$5 ERROR ROUTINE *** 149001 ON ERROR GOTO 19000 14910 GOSUB 14950 GOTO 14990 LOOK FOR & LF-CR-HULL END LINE SEQUENCE IF NO END OF LINE AND NO TEXT THEN RETURN ELSE SAVE THE REMAINDER LOOK FOR A CR-LF TO END BASIC LINE IF THE REMAINDER DOES NOT HAVE A CR-LF WE ARE NOT THRU READING THE BASIC LINE. GO CHECK FOR FOSSIBLE LOST END OF BASIC LINE. I RESET ERROR ROUTINE I GO SEND MESSAGE I GOTO TO SUB COMMAND EDIT EXIT L. D%=FNTD.LUG%(*? \$Sub ERROR 14950 *+SUB.ERRMSG\$+CR.LF\$.0%) SUB.BASIC.LINE.NO%=0% ! LOG MESSAGE SET BASIC LINE NUMBER≈ 0 FOR ERROR RETURN RETURN 14990 I END OF SUB COMMAND EDIT ROUTINE **** #Sub COMMAND BASIC LINE READER ROUTINE **** 16030 THIS ROUTINE IS ACTIVATED BY THE \$\$ ROUTINE BY SETTING THE SUB.SW% AND SENDING THE PK JOB A 'LISTNH o' VIA 10000. SUB.LIME#=LEFT(TO#,SUB.CRX+1%)+CHR\$(0%) TO4=RIGHT(TO#,SUB.CRX+3%) THE PURPOSE OF THIS ROUTINE IS TO READ THE BASIC LINE FROM THE PK JOB AND SAVE IT IN AN ARRAY (SUB.TEXT#). 8

I EXTRACT THE NEW LINE I STRIP THE LINE FROM THE PK TEXT

ELSE

10 GET SOME MORE

SUB.TEXT.REM\$=T0\$ GOTO 16010

RETURN.

SUB. TEXT. REMS=TOS

L.CRLF%=INSTR(1%,SUB.TEXT.REM\$,CR.LF\$) IF L.CRLF%=0% THEN 16380 ELSE

GOSUB 16500 SUB.SW%=0 GOTO 16010

GOTD 16390 IF SUB.TEXT%=0% SUB.LINE\$=LEFT(SUB.TEXT.REM\$,L.CRLF%+1%)

REJECT REQUEST IF CR-LF FOUND ON FIRST LINE, WE MAY NEVER SEE THE LF-CR-0 OF COULD LOSE TEXT ON FOLLOWING LINES. NOTE: NOW WE HAVE A NOEXTEND BASIC LINE WITH MORE THAN A

EASIC LINE WITH MORE TH CR-LF, GET THE LINE TO BE SAVED GO SAVE THE LAST LINE RESET THE SUB SWITCH RETURN

RESPONSE

16360 GOSUB 16500 GOTO 16320 . GO SAVE THE LINE ! REPEAT PROCESS UNTIL NO COMPLETE LINE ARE FOUND **** ROUTINE TO CHECK FOR NOTEXT PROBLEM IN NO EXTEND MODE **** & 143801 IF INSTR(1%, SUB. TEXT. REM\$, "Ready")=0% THEN 16010 16385 CHECK FOR READY PROMPT EXTEND LINE IN NO EXTEND MODE FALL THRU TO REJECT ROUTINE BECAUSE WE MISSED THE CR-LF BY SUBING AN EXTEND LINE IN NOEXTEND MODE 16389! *** ROUTINE TO REJECT REQUEST IF EXTEND LINE IN EXTEND MODE *** 16390 SUB. NOEX . FEEZ=12 SUB.SW%=0% GOTO 16010 SET ERROR SWITCH RESET SAVER SWITCH RETURN SO SUB ROUTINE CAN ABORT RERUEST 16499! **** TEXT SAVER SUBROUTINE **** 16500 SUB. TEXTZ=SUB. TEXTZ+12 SUB.TEXT\$(SUB.TEXT%)=SUB.LINE\$ RETURN BUMP COUNT OF LINES SAVED SAVE THE LINE IF ERL>14000 AND ERL<14999 THEN RESUME 14900 19030 ! TRAP \$S INTERPRETER ERRORS IF ERL=13660 THEN RESUME 13670 19031 ! TEST FOR VAL ERROR WHEN RESETTING BASIC LINE NUMBER 19032 IF ERR=28 THEN RESUME 4100 I TRAP CC AND ATTEMPT A RESTART 32767 END Patch to ATFK.BAS (V7.0-07B 08-JUL-80) line 16000: *H/16000<tab>!/V<cr> 16000 (tab>! & (cr) *H/DEF*/User tab>DEF* FNT0.LOG%(T0\$,T0%) &<cr> *AV<cr <tab>\ IF (NOT LOG.OFF%) THEN &<cr> #I<cr> <pr> <cr><cs>¥V<cr><ctab>\ IF (NOT LOG.0FF%) THEN &<cr> #2C/16007/V<cr> 16007 IF (NOT LOG.OFF%) THEN &<cr> #EX<cr> Checksum: 65395 Magnetic tape format: DOS/ODD:800 BPI Directory: Name .Ext RTCULL.DOC ATFKED.RNO Prot Date <155> 17-Sep-81 <155> 17-Sep-81 <155> 17-Sep-81 <155> 17-Sep-81 Size Prot MM: [2.1] (COPY #1) 51 9 (Complete article) (Section 1.0 thru 4.*) (ATFK Patch) ATPKPA, DOC ATPKED.BAS 38 <155> 17-Sep-81 (ATPKED.BAS append code) Total of 100 blocks in 4 files in MM:[2,1] Prot Date <155> 17-Sep-81 <155> 17-Sep-81 <155> 17-Sep-81 <155> 12-Sep-81 <155> 12-Sep-81 Size 51 9 Name .Ext RTCULL.DOC MM:[2,2] (COPH #2) ATPKED, RND ATPKED.BAS 38 <155> 17-Sep-81 Total of 100 blocks in 4 files in MM:[2:2] Grand total of 200 blocks in 8 files in MM:[*,*] AT LAST !!! The RSTS PROFESSIONAL will be published six times a year. February, April, June, August,

October, December

RSTSPROFESSIONALRSTSPROFESSIONALRSTSPROFESSIONALRSTSPROFESSIONALRSTSPROFESSIONALRSTSPROFE



NPI proves purchasing Disk Sub-systems does not have to be

HIGH RISK vs BIG SAVINGS

- Disk drives manufactured by the same vendor DEC buys them from
- No operating systems software changes
- One day installation
- 100% software transparency
- DEC's RM05-RM03-RM02 drivers are used
- 50% cost savings
- Single vendor maintenance in major cities
- Uses DEC's diagnostics
- Timely delivery
- DEC RP06-RP04-RM03 RM02 accepted on trade

NPI National Peripherals, Inc.

41 Chestnut Lane Westmont, IL 60559 (312) 325-9700

CIRCLE 62 ON READER CARD

R5TSPROFESSIONALR5TSPROFESSIONA



Send questions to: DEAR RSTS MAN. P.O. Box 361, Fort Washington, PA 19034.

DEAR RSTS MAN:

Contiguous Isn't Always Best, or How I Learned To Use UTILTY FLAG/NOCTG

It is sad that people quite often won't spend some time to save more. It's even sadder when, as I did, you spend the time and then have to waste time figuring out what went wrong.

Having just upgraded a RSTS system from 4 RK05's by adding an RM03 equivalent, I was delighted that the account copy program I'd written worked so fast. Better still, it created all contiquous output files - that has to be better, doesn't it?

Well, the answer is "not necessarily!" Everything worked beautifully until an invoicing program gave a protection violation on a CLOSE statement. Dim* memories awoke, and I remembered having the same problem with virtual arrays a couple of years ago.

The difficulty this time was with a disk file opened in mode 2 (append). So little information had been written to this file that there had been no need to write to disk from the buffer during program execution. The CLOSE, of course, caused exactly that to happen and RSTS was complaining in somewhat confusing style about my attempt to extend a contiguous file. Thank heavens for UTILTY! *No pun intended

Steve Holden

Dear Steve: No doubt RSTS can confound us! Try some strange protection codes, < 3 > for example, and try PIP/ZE.

DEAR RSTS MAN:

I have a large file and I would like to read through it looking for only certain records that match what I want. Should I use large record-sizes or what? It takes too long on my system with my normal programs. What can I do? **Overnight Batch**

Dear Overnight Batch: Come back to daylight. RSTS MAN has discovered SELECT*. This Evans, Griffiths and Hart product will fill your needs. It is a general utility that will extract records that fit your specifications from a file and write on output file of your specifications (sorted optionally) with unbelievable speed. It extracted records from a 1.000.000 (million) record file (20 byte records) in 20 cpu minutes on an 11/70. The 40,000 record 20,000 block master file can be scanned in 30 cpu seconds!

*ed: Look for a complete article on SELECT and its uses in the February issue.

FROM THE EDITORS . . . DAVE MALLERY ... continued from page 4

prints fox messages on KBO: after each subpass and next to RSTS, is the ultimate CPU and memory test.

TRAP to 114-a parity error in CACHE-randomly every twenty minutes or so, during EQKC. Trouble-right here in River City. At this point, we spent a week tearing through the memory boxes trying to find more trouble. Found and fixed a badly fraved Berg cable coming off the cache board-no cigar (no spares, either). It's amazing how blind you can be without a teacher.

The great awakening came when I finally opened my eyes and looked at the memory error registers in the legions of 'different' trap to 114's I had accumulated-they looked like this:

ERRREG:004020 or 04420 etc.

but always ended in 20. Looking at the processor book for the umppteenth time (reading it for the first time) I realized that the 20 bit meant an error in CACHE address group zero! And it was there every time!

You can disable cache (or half of it) by setting an few bits in a cache control register. I re-booted the diagnostic (EQKC), started it normally, quickly halted and turned off cache group zero. Success! 350 passes over a weekend with zero problems.

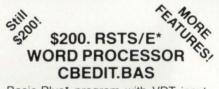
Now, all I have to do is get the cache board fixed. I stuck it into #105 to be sure I had the right board (I did) and off to the phone. Three out of four companies politely declined, the last one, a new firm in NYC, tried valiantly and very professionally for days with an on-board chip testing computer changed several chips, but couldn't fix it.

Now DEC offers depot service and will fix it for you for \$155 but takes at least 8 weeks. Not guite what I had in mind! So, biting the bullet, I paid \$1380 for a SWAP! (60% of list). So now, old #71 is back on service, and in service. I expect another six years of flawless operation.

Some Morals:

- 1) Old 70's are getting very cheap! If you are persistent and armed with cash, you can make some excellent deals. Buy with care-unless the machine is RUNNING and under current DEC service, they won't take it on service without a prelude of time and materials inspections that can be very expensive. Check with your Field Service manager before you buy!
- 2) Fixing old 70's yourself without spares can be very enlightening, educational and time consuming.

- 3) Depot service houses can fix most anything except a broken cache.
- 4) You will never get a deal like the above from a broker, but you can make the broker fix the machine.



Basic-Plus* program with VDT input, window edit and document save. Add, locate, global change, replace, delete, block move and file merge, etc. Crash and operator error recovery. Supports DEC, Hazletine and Mime standard VDT's. Others easy to add.

Fully formatted output (margins, justify, center, underscore, super-sub-script, headers, page numbers, etc.), to typewriter, line printer or disk. Bidirectional driver for Diablo-Xerox 16-17 series.

User's manual and source code included. 9-Track \$200. RKO5 \$260. ppd T. F. Hudgins & Associates, Inc. P.O. Box 10946, Houston, Texas 77018 Woods Martin 713/682-3651

*TM Digital Equipment Corporation

CIRCLE 9 ON READER CARD

DEC USERS: Stop and think. Our disk systems are the only viable storage alternative. . .



- More reliability. We'll provide great alternative solutions—like our Add-On, Non-Stop Reliability (ANSR) systems. . . or our 160 or 675 Mbyte, SMD-compatible Winchester storage drives-for added dependability and increased throughput.
- More flexibility and floor space. We've eliminated the problem of limited choice. Name Le it; you've got it. And our alternative solutions take up much less floor space.
- Faster delivery. We've built our solid reputation on guick delivery. Typically, we can give DEC users value-packed storage systems in 30 to 45 days ARO.
- Responsive service. We have service centers both in the U.S. and Europe. We offer 4-hour response to any call within a 50-mile radius of major U.S. cities. And 24 hours/ day, 7 days/week service contracts for all key metropolitan areas.
- Big money savings. For example, two RMO3s with a total capacity of 160 Mbytes cost 5 \$50,000. Three of our 160 Mbyte Winchester drives in support of a DEC VAX-11/780 give you 480 Mbytes for \$42,500.

For more information about the only viable alternative to DEC data storage systems, contact us today at any of the addresses listed below.



United States: 525 Oakmead Parkway, P.O. Box 425, Sunnyvale, CA 94086, (408) 732-1650, Telex 346-459. Europe: System Industries (Europe), System House, Guildford Road, Woking, Surrey, GU22 7QQ, England, (048 62) 5077, Telex 859124. California (714) 754-6555, (213) 557-0384; Colorado (303) 741-3502; Georgia (404) 955-2252; Illinois (312) 948-9330; Massachusetts (617) 695-4022; New Jersey (201) 839-8650; New York (516) 482-6082; New York Metro (212) 953-0315; Ohio (513) 771-0075; Seattle (206) 451-8791; Texas (713) 497-7224; Washington, D.C. (703) 734-9700; West Germany (06102) 5464/5; Sweden 08-63 62 74

CIRCLE 20 ON READER CARD

TIMER.BAS

By Michael H. Koplitz, Allis-Chalmers

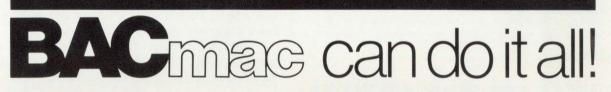
10	EXTEND		240	GOTO 270 IF REMOVE%(Y%) = 0%	ISKIP THIS ONE IT REMAINS IN TABLE			
	This program is designed to log out users who stay ir	C state for more	250	GOSUB 5000	REMOVE ENTRY FROM TABLE			
	than 10 minutes. It will not effect [1,*] account this program runs every 3 seconds so that it is p	possible that a	260	Y8 = Y8 - 18	BACK UP ONE SINCE TABLE			
1111	user might be able to be forced out even though t ^C state during the MINUTES.WAIT% minute interval 10000 to 1.				IS MOVED UP BY ONE IN 5000 ALSO NOTE THAT TABLE.COUNTER& DECREASED BY ONE.			
1	The only requirments of this program is that you hav Matrix functions are not neccessary, they are not	ve BASIC-PLUS.	270	NEXT Y%	CONTINUE LOOPING			
11	program. Note that this program does take up a job slot so yo		280	SLEEP 3%	ISLEEP 3 SECONDS BEFORE MAKING NEXT PASS.			
	to run it all the times.		290	GOTO 170	MAKE NEW PASS. NOTE THAT THE TABLE REMOVE IS ALREADY ZERO DUE TO			
	RUN TIMER			**************************************	LINE 230-270.			
11	How many minutes to wait ? XX		1/					
	Timer will be detaching in job slot ##		\! IMPORTANT NOTE: ONLY A ^C WILL REMOVE THE PROGRAM OUT OF ITS PROCESSING. \! A ^C TRAP IS UNNECESSARY BECAUSE THERE AREN'T ANY FILES OPEN. IT DOES \! NOT MATTER WHERE THE CODE FINISHES OUT. IF A ^C IS ENTERED WHILE IN					
$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$	Stoping the program:		\! THE	MIDDLE OF A SYS CALL, THE SYS CALL WILL I DRE THE PROGRAM RETURNS TO ^C Ready STATE.	BE COMPETED BY THE FP			
$\begin{pmatrix} 1\\ \\ 1\\ \\ 1 \end{pmatrix}$	 Attach to the job slot timer is running in. C the program. 		/!****	*****************	****************************			
	Author: M H Koplitz DP Analyst		1000!**	**************************************	DUT************************************			
$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	Allis-Chalmers HTD E. Berlin Road			This routine will search the job table, table and it is not in [°] C status (STATU:	S.JOB% >= 0) then the entry in			
	York, Pa. 17404		\! \!	the table will be flagged for deletion job is found and it is in ^C status the	(REMOVE%(Z%) = 1). If the time will be examined to			
	Date written: 05-Dec-80			determine whether the job has been in "(MINUTES.WAIT% minutes, if so send a mess	C status for more than sage and BYEF. If the job			
	Justification: The program will eliminate those user a drain on system resources. Those users who log out when they are finished with a session. The mo	in and don't log	\! \! \!	is not found in the table and it is in ' the table with the current time (TIME\$((C status, enter it into			
$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	constantly monitor these jobs.		1 *****	Note that accounts [1,*] are skipped.	*****			
	************	*****	1010	Y\$ = SYS(CHR\$(6%)+CHR\$(26%)+CHR\$(X%))				
30	DIM JOB.NUMBER% (32%), PROJ% (32%), PROG% (32%), ENTRY.TIME\$ (32%), REMOVE% (32%)		1010	19 - 515(CHK\$(06) +CHK\$(206) +CHK\$(X6))	SYS CALL TO GET THE JOB STATUS S% = 0%, REFER			
1	1DIMENSIO				TO PROGRAMMER MANUAL.			
40	PRINT "TIMER V1.0 ALLIS-CHALMERS HTD ";D 1PRINT BA	ATE\$(0) NNER	1020	TEST.PROJ% = ASCII(MID(Y\$,22%,1%)) GOTO 1230 IF TEST.PROJ% = 1%				
50	PRINT ISKIP LIN	E ON TERMINAL	1	TEST.KB% = ASCII(MID(X\$,4%,1%)) GOTO 1230 IF TEST.PROJ% AND 128%				
60	INPUT "How many minutes to wait ";MINUTES.WAIT% !GET MINU	TES TO WAIT	1	0010 1230 IT 1651. FROM RRD 1206	<pre>!PROJECT NUMBER. SKIP IF ACCOUNT [1,*] OR DETACHED.</pre>			
70	PRINT ISKIP A L	INE	1030	TEST.PROG% = ASCII(MID(Y\$,21%,1%))	IPROGRAMMER NUMBER			
80	PRINT "Timer will be detaching in job slot ";PEEK	(518)/2	1040	TEST.JOB.NUMBER% = ASCII(MID(Y\$,3%,1%))/	2%			
	FEEK(518	F DETACHING.) IS THE FIXED LOC- MONITOR FOR JOB	/		IJOB NUMBER			
	NUMBER T		1050	FOR 2% = 1% TO TABLE.COUNTER%	LOOP THROUGH TABLE			
90	X\$ = SYS(CHR\$(6%)+CHR\$(7%)) 1DETACH C		1060	GOTO 1080 IF TEST.PROJ% = PROJ%(AND TEST.PROG% = PROG%(2 AND TEST.JOB.NUMBER% = J	(8)			
100 110 120	<pre>:************************************</pre>		N	AND 1631.000.NOMBERS - C	IA MATCH IN THE TABLE. DO THIS WAY TO AVOID <> STATEMENTS.			
130 140	ENTRY.TIMES(X) = "" NEXT X		1070	GOTO 1190	SKIP THIS ENTRY IN TABLE			
145	TABLE.COUNTER% = 0 XS = SYS(CHR (6%) +CHR (-3%))		1070	0010 1170	NO MATCH WITH THIS JOB			
	! TABLE.	GET FROM MONITOR	1080	GOTO 1180 IF STATUS.JOB% >= 0	IGOTO SEQUENCE TO REMOVE THE ENTRY FROM THE TABLE SINCE THE JOB IS OUT OF [°] C			
150! \! \!	JOB.NUMBER% KEEPS THE JOB NUMBER OF JOB IN °C PROJ% KEEPS THE PROJECT NUMBER OF THE JO	STATUS DB IN JOB.NUMBER%	1090	HOURS = LEFT(ENTRY.TIME\$(2%),2%)	STATUS			
NI.	PROG% KEEPS THE PROGRAMMER NUMBER OF THE TIMES THE TIME WHEN THE ^C STATUS WAS R	ECORDED	1	HOURIS = LEFT(TIMES(0),2%)	IGET HOURS			
NI .	REMOVE% INITIALLY ZERO WHEN JOB SCAN BEGIN JOB SCAN IT WILL EQUAL 1 IF THE .	JOB IS TO BE		**************************************	ED ON THE 24 HOUR CLOCK			
NI .	REMOVED FROM THE TABLE. JOB IS RI TABLE IF IT IS LOGGED OUT OR OUT JOB.MAX% MAXIMUM NUMBER OF JOBS ALLOWED DUB	OF °C STATE.	1	IF YOU USE THE AM/PM CLO A CONVESION ROUTINE TO C	HANGE AM/PM TO 24 HOUR.			
1	JCB.MAX% MAXIMUM NUMBER OF JOBS ALLOWED DU TIMESHARING.		/:!****	***************************************	**********			
160	ON ERROR GOTO 32000 ISET UP ER		1100	MINUTES\$ = RIGHT(ENTRY.TIME\$(2%) MINUTES1\$ = RIGHT(TIME\$(0),4%)	,4%)			
	1*************************************		5		IGET THE MINUTES			
180	FOR X% = 1% TO JOB.MAX% ITHE VALUE STATEMENT		1110	TIME.CREATED% = VAL(HOUR\$) * 60% TIME.NOW% = VAL(HOUR1\$) * 60% + V	+ VAL(MINUTES\$) /AL(MINUTES1\$) !CALCULATE TIME IN MINUTES			
	IS JOB AFT	TER SYSTEM JOBS,	1120	IF HOUR\$ = "23" AND HOUR1\$ = "00"				
185	GOTO 220 IF X% = PEEK(518)/2 !SKIP THIS		1	THEN TIME.NOWS = TIME.NOW				
190	X\$ = SYS(CHR\$(6%)+CHR\$(26%)+CHR\$(X%)+CHR\$(MAKE 00 HOUR EQUAL TO 24 BECAUSE YOU ARE SCANNING BEFORE MIDNIGHT AND AFTER			
200	<pre>STATUS.JOB% = CVTS%(MID(X\$,21%,2%)) IGET STATU</pre>	IS OF JOB,	1130	TINE.CHANGE% = TIME.NOW% - TIME.C	MIDNIGHT CREATED%			
		°C STATUS			HOW MANY MINUTES HAVE GONE BY IN °C STATUS.			
210	GOSUB 1000 ITABLE CHE		1140	GOTO 1230 IF TIME.CHANGE% < MINU	res.wait% Skip if not in ^c status			
220	NEXT X* ICONTINUE GOGUB 6000 ICLEAN UP				LONG ENOUGH. LEAVE ROUTINE.			
225 230	FOP Y* = 1 TO TABLE.COUNTER* !LOOP THRC	UGH TABLE LOOKING	1150	KES = MID(YS,4%,1%)	GET THE KB NUMBER FOR			
	FOR REMOV	$E \mathcal{K} (Y \mathcal{K}) = 1$			SYS CALL FORCE.			

ember	1981		

Dec

page 39

PSTSPROF	ESSIONALRSTSPROFESSIONALRSTSPROFESSIONALRSTSP	BOFESSIONAL BSTSPBOFESSIONAL BSTSPBOFES	SIONAL BSTSE	PROFESSIONAL PSTSPROFESSIONAL PSTSPROFESSIONAL	Page 33
no i or nort	SSIONALISTSPROFESSIONALISTSPROFESSIONALISTSP	NOT ESSICITATING IS FROM ESSICITATING IS FROM	SIUNALISISI	NOTESSIONALASTSCHOLESSIONALASTSCHOLESSIONAL	1313F NOTESSIONALISTSF NOTESSIONALISTSF
1160	Q\$ = SYS(CHR\$(6%)+CHR\$(-5%)+KB + "? Job in Ready status (" minutes - forced" + " logout"+CHR\$(13%)+CHR\$	over "+NUM1\$(MINUTES.WAIT%)+ \$(10%)) 1PRINT MESSAGE ABOUT LOGOUT		PROJ% (TABLE, COUNTER%), PROG%(TABLE, COUNTER%), REMOVE%(TABLE, COUNTER%)) = 0% ENTRY.TIME\$(TABLE, COUNTER%) = ""	
		ON TERMINAL	/		ICLEAR OUT LAST ENTRY IN TABLE
1170	Q1\$ = SYS(CHR\$(6%) +CHR\$(-4%) +KE	3\$+"BYE F"+CHR\$(10%)+CHR\$(13%)) !FORCE A BYEF AT THE TERMINAL	5060	TABLE.COUNTER% = TABLE.COUNTER% - 1%	IDECREMENT TABLE SIZE
1180	REMOVE%(2%) = 1%	IFLAG FOR DELETION	5070	RETURN	!ALL DONE
1185	GOTO 1230		6000 1	***************************************	*********
1		LENTER A ONE IN THE TABLE GOTO END OF ROUTINE		This routine checks to see if the jobs logged in	s in the table are still
1190	NEXT Z%	ICONTINUE LOOPING	/!****	*************	*********
1200	GOTO 1230 IF STATUS.JOB% >= 0	LENTRY NOT IN TABLE	6010	FOR Z2% = 1% TO TABLE.COUNTER%	LOOP THROUGH TABLE
		AND NOT IN ^C STATUS SO SKIP	6020	GOTO 6060 IF REMOVE%(Z2%) = 1%	WHO CARES IT WILL BE REMOVED ANYWAY LATER
1210	TABLE.COUNTER% = TABLE.COUNTER% + 1%	IINCREASE TABLE BY ONE	6030	F\$ = SYS(CHR\$(6%)+CHR\$(26%)+C1	IRS (TOR NUMBERS (725))
1220	JOB.NUMBER%(TABLE.COUNTER%) = TEST.JOB. PROJ%(TABLE.COUNTER%) = TEST.PROJ%	NUMBER%	1	10 - 515(chk0(06)+chk0(206)+ch	IGET JOB STATUS
1	<pre>PROG%(TABLE.COUNTER%) = TEST.PROG% ENTRY.TIME\$(TABLE.COUNTER%) = TIME\$(0) REMOVE%(TABLE.COUNTER%) = 0%</pre>	IENTER THE JOB INTO THE TABLE	6040	GOTO 6060 IF PROJ&(Z2%) = ASC AND PROG%(Z2%) = ASCII	
		WITH THE CURRENT TIME.	6050	REMOVE%(22%) = 1%	THIS TABLE ENTRY NO LONGER
1230	RETURN	SUBROUTINE FINISHED	6060	NEXT Z28	
5000 1*	****	*****			
\! \!	This routine removes the entry from the		6070	RETURN	
11	**************************************		320001	**************************************	******
			1		
5010	JOB.NUMBER%(Y%), PROJ%(Y%), PROG%(Y%), REM ENTRY.TIMES(Y%) = ""	10VE%(Y%) = 0%	/1****	******	*******
Ň		ICLEAR OUT OF TABLE	32010	IF ERL = 190 THEN RESUME 220	1JOB DOES NOT EXIST
5020	FOR Q% = Y% TO TABLE.COUNTER% - 1%	LOOP THROUGH REMAINDER OF TABLE AND MOVE UP ENTIES	32015	IF ERL = 1010 THEN RESUME 220	JOB DOES NOT EXIST ANY MORE
5030	JOB.NUMBER%(Q%) = JOB.NUMBER%(Q PROJ%(Q%) = PROJ%(Q% + 1%)	(% + 1%)	32020	IF ERL = 1020 THEN 1230	JOB LOGGED AT THIS TIME
	PROG%(Q%) = PROG%(Q% + 1%) REMOVE%(Q%) = REMOVE%(Q% + 1%) ENTRY.TIME\$(Q%) = ENTRY.TIME\$(Q	(% + 1%) !MOVE THE TABLE ENTRY UP 1	32030	IF ERL = 6030 THEN REMOVE%(22%) = 1% RESUME 6060	IJOB IS LOGGED OUT, ERROR IN GETTING SYS CALL STATUS THEREFORE REMOVE ENTRY
5040	NEXT 0%	ILOOP TILL DONE	32040	ON ERROR GOTO 0	INDICIONE REMOVE ENTRY
5050	JOB.NUMBER% (TABLE, COUNTER%),		32767	END	IALL DONE



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.

Please write for more information

CIRCLE 47 ON READER CARD

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

BLINK: A BASIC PLUS PREPROCESSOR

By Steve Holden, Manchester University, Dept. of Computer Science, Manchester, England

ABSTRACT

A preprocessor is described for use in a BASIC programming environment. The available directives, and some techniques which can be used to overcome weaknesses of BASIC PLUS, are presented. Experience with the preprocessor as a tool for producing commercial software is described.

INTRODUCTION

About eighteen months ago my company was charged with developing the software for a small PDP-11 bureau, using the RSTS/E operating system.

With an 11/34, and only a small development team, we had to come up with a way of increasing the productivity of our programmers. We decided to base our development effort on BASIC PLUS, which offers quite fast program development and a relatively easy upgrade to the compiled language BASIC PLUS 2. This could later be used to make more effective use of system resources.

The major software tool has been a preprocessor, which accepts a superset of BASIC PLUS.

What BLINK Does

The BLINK preprocessor reads a source file which looks like BASIC PLUS with certain things added (BLINK directive lines, labels, set symbols), and certain others left out (line numbers).

From this source file BLINK produces a BASIC PLUS file which can be loaded and compiled in the usual way.

Reasons for Development

BLINK is a response to some of the problems of programming in BASIC PLUS. Whilst the base language is adequate for expressing algorithms, there are few facilities which encourage the programmer to develop suites of programs in a neatly modular way.

The BLINK project was started because our standards, based on those published by DECUS, called for certain types of code to be allocated line numbers within certain ranges. We found it difficult to move routines from applications libraries into our standard libraries, which both started out as APPENDable files.

An obvious solution was to get the computer to enforce the standards for us. BLINK allows us to delegate to the system those tasks which we find least tasteful and most time-consuming.

The BLINK Language

BLINK is mostly BASIC PLUS. Indeed, as an aid to migration we have ensured that BASIC PLUS will pass through the preprocessor without change.

A line of BLINK may be in one of six types.

Numbered Lines — The line number is used to set BLINK's program counter, so that subsequently generated line numbers will be higher in sequence and thus follow on in the usual way.

Labelled Lines — A label is recognised when a normal extend-mode identifier begins in column one. It must be followed by a colon, which makes it look much like a label in more conventional programming languages.

The label is entered into the label table, a line number is allocated to it by incrementing the program counter, and a new line of output code is commenced.

Comments — To encourage liberal use of comment as a documentation aid, any line beginning with an exclamation mark in column one is considered to be a comment, and completely ignored by the BLINK preprocessor.

Directives — All lines beginning with a period are considered to be requests for the preprocessor to take some special action. The various directives are discussed at some length below.

Code Lines — Any line which starts with a blank or a tab is considered to be code. Such lines are passed through to the output stream after certain symbol substitutions have been made.

Blank Lines — These are not considered significant when they precede a directive line, allowing the conscientious programmer to improve the readability of his code. For simplicity of processing, we assume that all code continuation is indicated by ampersands as in BASIC PLUS 2 and the extend-mode convention.

A blank line thus triggers incrementation of the program counter and the start of a new output statement. They are often used to delimit the scope of a multistatement THEN or ELSE group.

With LSI 11/PDP 11 Software Compatible Disc/Tape Controllers Offering Single Board Low Power µP Based **Design and Low Cost... Plus Many Other Good Reasons!**

The reasons start with DILOG'S (Distributed Logic Corp's.) full time engineering and design staff. Not outside suppliers. That means when you contact DILOG for product selection or after sale service, you'll get "first hand" assistance...along with years of experience manufacturing µP based controllers that interface with DEC 11 CPUs.

The intelligent products you'll discuss all utilize common proprietary architecture and DILOG automated design techniques-products with exceptional reliability and cost efficiency...mostly available from stock. And

when you plug a DILOG controller into your DEC CPU it's ready-to-run because it's fully operating system software compatible.

These high performance data storage interface products also feature • minimum bus/space requirements • up to 60% less power • 10 to 50% lower cost • automatic self-test... and numerous other features for easy system integration.

Consult the DILOG/disc-tape compatibility table for your needs. Then ask for detailed data on existing, or future products from DILOG...#1 in single board DEC 11 compatible disc/tape controllers. Distributed Logic Corp., 12800-G Garden Grove Blvd., Garden Grove, CA 92643, Phone: (714) 534-8950 • 64-A White Street, Red Bank. New Jersey 07701 Phone: (201) 530-0044

MAGNETIC TAPE		DISC					
1/2" REEL-TO-REEL STD. & STREAMER	2315/5440/RK05 CARTRIDGE CLASS	CMD CARTRIDGE MODULE	SMD STORAGE MODULE	WINCHESTER 5¼", 8" OR 14"	1/4 " TAPE CARTRIDGE	FLOPPY DISC DRIVE	
STD. & STREAMER AMPEX CIPHER CONTROL DATA DIGI-DATA KENNEDY MICRODATA PERTEC TANDBERG DATA (IDT) WANGCO TDX	AMPEX CAELUS CENTURY DATA CONTROL DATA DEC DIABLO IOMEC MICRODATA PERTEC WANGCO WESTERN DYNEX DRI	AMPEX CONTROL DATA	AMPEX CENTURY DATA CONTROL DATA BALL COMPUTER MITSUBISHI	BASF CONTROL DATA FUJITSU KENNEDY MEMOREX PRIAM SHUGART SEAGATE QUANTUM IMI	DEI KENNEDY PEREX QUANTEX	BASF DECITEK MICROPOLIS PERTEC REMEX SHUGART	
	DRI	and the second second	Ann	CAPACITIES .25 TO 30	00 MB		
rademark Digital Equipment Co	orp.				CIRCLE 15 ON READ		

DISC/TAPE DRIVE MANUFACTURER COMPATIBILITY CHART

BLINK SYMBOLS

The preprocessor handles two types of symbols not normally a part of the BASIC PLUS language.

Labels are substitute for line numbers, which we feel 'harden' BASIC PLUS code, and make it less likely that a piece of code will be usable elsewhere.

A reference to a label in the code is preceded by the commercial at sign (@) because this makes them easy to detect by the preprocessor and also flags them to the eye of the human reading the code.

It is interesting to note that the "at" sign is one of the few with no syntactic significance to the BASIC PLUS runtime system. It was chosen in preference to the others, after several false starts, largely because it is available without use of the shift key on most of our locally available keyboards.

The second type of symbol is the so-called 'set symbol,' which may hold an arbitrary string. Several of the BLINK directives are concerned with setting and testing the value of these symbols.

They can thus be used

- to control the inclusion or exclusion of a given piece of code from the source file, allowing systems to be configured by feature selection.
- to generate different literal strings or variable names during different preprocessing runs, allowing the same source to produce code which is tailored to the circumstances.
- to modify the actions of library routines, or the code that such routines generate, giving a more flexible applications library without undue duplication of code.

The value of a set symbol is obtained in the program text by flagging the reference in the way shown below.

PRINT "@/TYPE/Ledger"

The above code fragment could result in the printing of "Sales Ledger", "Purchase Ledger" or "Nominal Ledger" according to the value of the set symbol TYPE when the code is processed.

To assist in such configuration a means is provided of setting symbols in the CCL command which triggers a BLINK preprocessing run.

It is interesting to note the parallels between programming in BLINK and programming in an assembly language followed by link-loading.

In both cases, the code produced from symbolic sources is absolute, although BASIC has the advantage that insertion of new code is much easier than it is in machine code.

In both cases the programmer's task is eased by allowing him to make use of pre-written libraries, eliminating duplication of effort.

Furthermore, it is the task of assigning fixed numbers (addresses in assembly language, line numbers in BLINK) which prompted the development effort.

This parallel was the main spur to the BLINK project, and the facilities of MACRO-11 and the task builder were used a model when various enhancements were made to BLINK. It is a sad reflection on the state of commercial programming that many of the features for long taken for granted remain unavailable in one of the most popular languages. Naturally the introduction of BASIC PLUS 2 has changed the situation somewhat, but we feel that even in a PLUS 2 programming environment many of the configuration abilities of BLINK would continue to be useful.

BLINK DIRECTIVES

Apart from the symbol substitution features mentioned above, BLINK enriches the base language by processing directive lines.

A BLINK directive is flagged with a period in column one, followed by a keyword to specify the required action, and optionally a list of arguments to modify the action.

There are several types of directive, and they are split for the purposes of discussion into several different classes.

Code Sharing Directives

One of the best ways to improve programmer productivity is to make it easier to use someone else's code than to write a routine yourself.

BLINK has facilities to allow a section of code from another file to be inserted in line, which is similar to the action of an APPEND command. It also has library searching facilities, which help to reduce the number of files on the disk, and reduces file processing overhead during preprocessing.

.INCLUDE file

takes the text contained in the named file (with default of SY:(SELF).SRC) and processes it as though it replaced the INCLUDE directive in the source program. INCLUDE directives may be nested up to a depth of nine, although in practice we have not yet felt the need to go beyond two or three levels.

LIBRARY module

adds the given module name to the list of modules for which library search must be performed when the end of the toplevel source file is encountered.

SEARCH file

adds the named file (with default SY :(SELF).LIB) to the list of files to be searched for requested modules during library processing. The libraries are searched in the order of the SEARCH directives met during the source scan.

.NEEDS module

allows a library module to cause an error if some other required module was not requested.

Symbol Setting Directives

The set symbols are translated as having the null value if no value is assigned in the CCL command or during source

Pascal POWER for your RSTS Machine



LEAP INTO THE 80's with Accounting Software in Pascal. The TBS accounting systems, in use internationally for the past 2 years, are now available for the RSTS user. The systems are designed to fulfill your accounting needs for the years to come, using the speed and flexibility of RSTS and Pascal.

FIVE INTEGRATED FINANCIAL PACKAGES: A/R, G/L, A/P, Payroll, and Order Entry with Billing, Inventory

Control, and Sales Analysis.

- INTERACTIVE
- EASY-TO-USE
- COMPREHENSIVE
- MENU-DRIVEN
- VERSATILE
- WELL DOCUMENTED

- EXTENSIVE REPORTING CAPABILITIES
- CUSTOMIZATION SERVICES
- SOURCE CODE AVAILABLE HIGHLY PARAMETERIZED FOR
 - **RAPID MODIFICATIONS**

PASCAL DEVELOPMENT SYSTEM: A complete multi-user ISAM-type file handler and an extensive package of external procedures and utilities, designed to greatly reduce software development time. Standard maintenance and print programs can be created in less than an hour.

Pascal programs are easily maintained, updated, and modified. The systems are also available on RT-11, TSX-Plus, and RSX-11M. A UNIX and VAX-VMS version will be available soon. For pricing information, product descriptions, or demos, contact Theta Business Systems or our U.S. Distributor, Secteur Corporation.



SECTEUR CORPORATION

BAINBRIDGE PROFESSIONAL BLDG., SUITE 103 BAINBRIDGE ISLAND, WA 98110 (206) 842-5612



THETA BUSINESS SYSTEMS

1110 SONORA AVENUE, SUITE 106 GLENDALE, CA 91201 (213) 242-7981 or 245-0917

CIRCLE 63 ON READER CARD

processing. Thus a reference to an undefined set symbol is not considered to be an error.

.SET symbol = value

assigns the value on the right of the equals sign to the symbol named on the left. The assignment is unconditional.

.DSET symbol = value

acts like a SET, except that the assignment is only made if the symbol currently has the null value. This allows an IN-CLUDED or library text to assert default values for its generation parameters.

.ASK symbol

prompts for the value from the terminal if the symbol currently has the null value. This is a useful directive when generating a highly-configured piece of software, when there are two options open to the programmer.

He may generate the file as it stands, when he will be prompted at the terminal for the system generation parameters.

Alternatively, he may write a module which SETs some or all of the system generation parameters and then IN-CLUDEs the source file. Because of the null action when the symbol is already defined, he will then not be prompted for the parameters he has set when he generates this higherlevel module.

Conditional Assembly Directives

When writing turnkey software it is quite often desirable to include or exclude certain sections of code according to whether particular features are required (and will be paid for!).

BLINK has facilities to optionally skip sections of the source code, depending on the value of one or more arguments to the directives.

.ENDIF

delimits the scope of all conditional assembly directives given below. If conditional skipping is not taking place then it has no action.

.IFB string

.IFNB string

assemble the lines up to the next ENDIF if the remainder of the directive line is (or is not) blank. These directives are most often used to test the value of a set symbol, as in

.IFNB @/TEST/

where the following code is assembled only if the set symbol TEST has a non-null value.

.IFEQ string1,string2

.IFNE string1,string2

continue processing if the two strings are (or are not) identical. This is useful where one or more options exist during system generation, as in .IFEQ @/TYPE/,P (Purchase ledger code) .ENDIF .IFEQ @/TYPE/,S (Sales ledger code) .ENDIF

RSTSPROFESSIONALRSTSPROFESSIONA

At present there are no facilities to nest the conditional directives, although this is an obvious extension to the current language. Since an ENDIF has no action if skipping is not taking place, however, as long as the scope of two conditionals ends at the same place such an action can be simulated in a way which will be upwards-compatible with nesting when it is implemented. We can write after the above code

.IFNE @/TYPE/,P .IFNE @/TYPE/,S .PRINT ?Invalid ledger type @/TYPE/ .ENDIF .ENDIF

Since the first ENDIF delimits the scope of both conditionals, skipping can never be taking place when the second one is encountered, and it will be ignored. When nesting is implemented, however, it will delimit the scope of the second IFNE directive.

Program Section Directive

Many installations, ours included, have adopted standards which call for restrictions on the line numbers which may be used for certain types of code.

It is common, for example, to put array dimensions in at 900-998, initialisation code at lines 1000-1999, and so on.

Whilst BLINK has to a large degree removed the need to consider line numbers, it is still useful to know roughly whereabouts in the program a piece of code will be found — during interactive debugging, for example.

Furthermore, it often happens that a single source module must ensure that certain variables are preset during initialisation, or that a particular error at a certain line is catered for in the error handler.

This is catered for by splitting the range of permissible line numbers into named program sections, or PSECTs, which will be a familiar concept to the MACRO-11 programmer. Each PSECT has its own program counter, and so BLINK will resume output at the next available line number within that section when it is resumed.

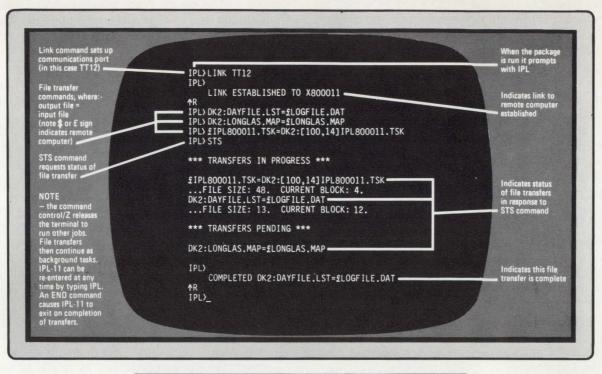
.PSECT {section-name} {,LD:n} 1,HZ:m

is the directive which causes the preprocessor to switch between different sections. We will deal with the most complex case first, where all arguments are specified.

.PSECT INIT, LD: 1000, HI: 1998

declares a program section, with limits on the line numbers which may be generated from within it. If a PSECT named

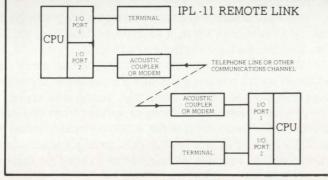
Do you own a PDP-11? Do you need to move files between PDP-11's?



XOREN IPL-11

The simple way to transfer PDP-11 files

XOREN IPL-11 is a software package which enables files to be transferred over a communications link between two DEC computers each of which may be a PDP-11, an LSI-11 or a VAX-11. Using the package, files can be transferred over a telephone line or a direct line.



- INTERFACE HARDARE -

No special interface hardware is required other than (in the case of remote computers) modems or acoustic couplers. Package operates via standard DEC terminal interface cards-DL11, DZ11, DH11, etc.

- DATA INTEGRITY -

CRC checking by software to CCITT recommendation V41. Re-transmission of bad blocks is used for error recovery.



Note:- Xoren Computing is currently setting up a distributor network for XOREN IPL-11 in the US. For further information contact Xoren Computing Ltd direct. - TRANSMISSION -

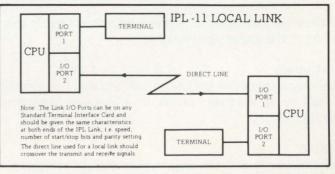
Asynchronous transmission with selectable speeds up to 9600 baud.

- OPERATING SYSTEMS -

Versions of this package are available now to run under RSX-11M, RT-11, RSTS/E, (and equivalent versions of CTS - 300 and CTS - 500), IAS and VAX/VMS in RSX-11M compatibility mode.

- LICENCE -

The package is normally supplied under a 5-year licence. A separate licence is required for each combination of CPU and operating system under which IPL-11 is to run. The two (5-year) licences required to link two CPU's cost \$1350.00 each, whichever pair of operating systems is specified. For larger orders a system of discounts is applied.



Xoren Computing Ltd. 28 Maddox Street, London W1R 9PF England Telephone (01) 629 5932 Specialists in:-Minicomputer Software Microcomputer Software Data Communications Interprocessor Links INIT already exists then this attempt to specify new limits is considered an error.

Most often, this form is used in a module which is IN-CLUDEd by a programmer almost at the beginning of his source code. If he does this, the PSECTS will be set up for him, and he does not need to bother about establishing his own PSECTS.

.PSECT INIT

causes the preprocessor to switch back into the INIT section, resuming code output where it left off when it was switched out. Thus, since BLINK normally generates code with increments of ten between line numbers, if the last line generated in the INIT section was 1050, it will recommence output at line 1060 (unless a line number appears to reset the program counter).

Lastly, the PSECT directive may be used without any arguments. This is principally convenient for INCLUDEd and library modules, as it switches into the PSECT which was in force when the module text was commenced.

One particularly useful application of the PSECT directive has been in error handling. Our standard program skeletons all set up a PSECT named ERRORS, and included initialisation code which traps errors to the start of this section.

Any INCLUDED module can specify private error trapping with the sequence

.PSECT ERRORS

IF ERL = $@$ LAB% AND ERR = 99%	&
THEN RESUME @LAB2	&

.PSECT

This obviates the necessity for routines to specify a private error trap, which we have found to be a fruitful source of errors in the past. Note also that, since label references are translated into line numbers by BLINK, it is legal to use them to specify an integer error line number.

Miscellaneous Directives

There are two other directives which do not easily fall into one of the above classifications.

.PRINT string

prints the remainder of the line on the terminal. It is useful for identifying selected options in a complex program generation.

.ENDFILE

is used to specify that the preprocessor skip the remainder of the file or library module it is currently processing. This is rarely used (although BLINK generates on internally when it actually encounters the end of any source file!), but can reduce processing time if an error is detected in the generation parameters.

PROGRAMMING TECHNIQUES

RSTSPROFESSIONALRSTSPROFESSIONA

It is difficult to present the features of the BLINK language in a coherent way given the space limitations imposed on this paper. There are many techniques which we have found lead to increased productivity by sharing code or reducing debugging time. In particular, we feel that the major advantages of the BLINK preprocessor are:

- A standard program skeleton can be written for each type of program (file create, report, et cetera), and these skeletons can be used by programmers with insufficient experience to define a sensible program structure for themselves. Use of these skeletons allows one programmer to understand another's code much more readily.
- The standard library, with disk handling routines and echo-control terminal input functions are available by writing a single directive for each module.
- When an error is detected in a standard module, it only need be corrected once.
- Since the standard modules are effectively hidden at source level, program listings are both smaller and more comprehensible. Also, we are encouraged to treat the standard routines as 'black boxes', and use only the interface descriptions rather than reading the code.
- Because BLINK encourages a modular approach to programming, programmers tend to structure their own code in a sensible way.

Without exaggeration, we claim that programming in BLINK is more enjoyable — although this is obviously subjective opinon.

We have traded the ability to develop programs on-line for the flexibility of using a higher-level language, and we feel that the exchange is in our favour.

We can write programs which have virtual array sizes adjusted to the requirements of each customer (by setting the dimensions in set symbols), thus saving disk space.

We can generate multiple programs from the same source code, using conditional assembly and putting IN-CLUDEd file names into set symbols, saving a maintenance load as well as keeping the number of source files down to more manageable proportions.

USER EXPERIENCE

BLINK has been used to produce several fairly complex sets of applications programs, now all in use on a timeshared PDP-11/34 in parallel with further program development.

Sales and Purchase Ledgers

This comprises about twenty programs, plus some associated file maintenance utilities. With the exception of



MENU THE MISSING LINK.

MENU is an applications development aid that provides the DEC RSTS software developer with a powerful process control device.

MENU easily generates menus that guide the application user into selected programs based on his security level.

MENU supplements RSTS security by allowing multilevel access capabilities within an individual account. Separate project-level control minimizes System Manager interaction for system level security.

MENU is driven by simple text files which determine extent of program control, type and level of security, screen displays, and presentation of on-line '/HELP' information.

MENU provides a separate Run Time System to prevent unauthorized access to RSTS 'ready state' resources. MENU installs in minutes and requires no software modifications.

MENU provides a common interface for all your users and application needs.

For information on MENU, please give us a call. We would be delighted to show you the missing link.



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

> 60 DAY TRIAL FOR \$5000

DEC and RSTS are registered trademarks of Digital Equipment Corporation.

programs concerned with producing cheques and remittance advices (unnecessary in a sales ledger), all programs are produced in two versions from a common source.

It is hard to estimate the amount of effort this has saved, but it has definitely cut the load of maintaining the system, which is in daily use by up to eight bureau clients.

Magnetic Tape Archiving System

The principal feature of this system is an index, held in virtual arrays, specifying which files were dumped to which tapes on which dates. It would obviously be wasteful to make the file big enough to cope with a thousand files when only a couple of hundred were to be archived.

A source file specifying the size of all these arrays is IN-CLUDEd in each program, allowing a simple systems generation procedure for turnkey clients requiring the package. At the same time they are customised as to client name.

Contract Costing System

This system was the first to provide integration between the sales and purchase ledgers, and a payroll system written outside our company in straight BASIC PLUS.

As purchase, sales and payroll details are entered they are also used to update a cost ledger maintained for all the client's contracts.

This application has its own source libraries, which contain routines to access hash-indexed files. They proved to be so successful that a more general version of them is shortly to be incorporated into the systems library.

General Libraries

These libraries are used by all applications suites. When turnkey packages are required for an outside customer it is a simple matter, for example, to update the routines concerned with

update would automatically be incorporated into all programs as they were generated.

Some Problems

The major current problem is that of sharing the development machine with an interactive bureau service. BLINK is written in BLINK, and can place quite a heavy load on a PDP-11/34 with storage module drives. Fortunately, the major development effort on the ledger packages is now over, and amendments can usually be made by editing the sources during the day and submitting an overnight batch job to regenerate the suite.

We feel, nevertheless, that BLINK could be usefully rewritten in some compiled language, although no statistics are available to indicate whether the generation process is disk- or CPU-bound.

BASIC PLUS 2 is the obvious candidate, since the generation process will work just as well for that as for BASIC PLUS. String handling still imposes quite a heavy overhead in PLUS 2, however, and for this reason we feel that MACRO-11 or PASCAL might give better performance.

At the same time, it would be useful to incorporate at least some lexical and syntactic analysis of the program. This would allow the preprocessor to decide for itself which functions had no definitions in the source programs, and should therefore be sought in the libraries.

It is somewhat annoying to go through the whole process of program generation only to be informed by the BASIC run-time system of syntax errors or undefined function calls introduced by the last edit.

Eventually it may be possible to produce a true compiler for BLINK, which output object modules for processing by TKB or LINK. With the increased usability of VAX BASIC and BASIC PLUS 2, however, the major advantage of this would

> be portability of applications software.

In this respect, a compiler which output code to run under RT11 would be especially useful in allowing migration of applications packages down to the LSI-11 based systems.

With proper design it should even be possible to transfer the language to completely different series of machines, as

visual display addressing and output and regenerate the programs to the new requirement.

Applications Libraries

As an example of the versatility of these library modules, should a turnkey customer require tight formatting on customer account references, only one library module need be changed. This



Cartoon by Kevin Macey, Submitted by Peter Dick, Silver Programs, London

has happened with BCPL and SPITBOL, for example. In the long run the user community could be expected to benefit although, of course, I am sure that Digital Equipment will continue to provide a high level of portability in their own software.

Code Samples

To give a small idea of the flavour of BLINK, a section of the preprocessor itself is shown. We hope this gives the reader some impression of the improved style which the BLINK programmer can assume.

.DSET T			
.DSET M	IXTES=100		
.PSECT	DIMS		
	TESZ=@/TES/		8
	<pre>\ DIM TEL%(@/MXTES/), STAE1*(@/MXTES/),</pre>	STABS(@/MXTES/)	8
	\ NX1%=0/16S/		*
.PSECT			
	DEF FNSN0%(X\$)		8
	<pre>\ SLOT%=(CV1\$%(X\$)+CV1\$%(RIGHT(X\$+LEN(X\$)))</pre>)-1%))) AND 32767%	8
	x SED12 SEU12-(SE012/18S2)*18S2		R
	<pre>\ WHILE IFL%(SLOT%) \ GOTO @SNU.ND IF STARI\$(SLUT%)=X\$</pre>		8
	\times 5LOTZ=1ELZ(SLOTZ)		R
	NEXT		8
	N GOTO 32767 IF FNEP%("SET SYMBOL TABLE	OVERFLOW	8
	IF NXTZ>=@/MXTES/		8
	\ STAB1\$(SL01%)≈X\$		R
	N TBLZ(SLOTZ)=NXTZ		×
	\ NXTZ=NXTZ+1Z		8
SNO. ND:	ENSN07=SLOTZ		8
	N ENEND		8
		I SELECT PSECT N	
		I ALWAYS RETURN PS NO.	
		I IF CHANGING PSECIS	8
		I AND LUAD NEW ONE	
		I SAVE ULD HIGH	8
		I AND LUAD NEW ONE	8
		I SAVE OLD LOW	8
	\ L0%=L0%(N%)	+ FIP LUND NEW DNE	8
	FNEND	I	8
	DEF ENESNO%(X#)	I FIND PSECT NUMBER	8
	N GOTO PES.ND IF PSNAM\$ (JUNK%)=X\$	OUT IF FOUND IT	8
	\ GOTO @F3.ND IF PSHAM\$(JUNKZ)=X\$ OR PSNAM\$(JUNKZ)=** FOR JUNKZ=1Z TO 252	OR NOT THERE	ĸ
	FOR JUNKZ=1Z TO 25%	SCANNING ALL NAMES	8
	N GOTO 32767% IF FNEF%(*:PSECT TABLE FULL	L (FNPSN02)*)	8
PS.ND:	FNPSNU%=JUNK%	SET RESULT	8
	1 FNEND		8

SUMMARY

BLINK is a first attempt to produce some much-needed support for the BASIC PLUS programmer. It has proved itself to be valuable in a commercial programming environment, although a more efficient implementation would be useful.

It also encourages the programmer to break down his programs into functional modules, since in this way he gains maximum benefit from the preprocessor.

We would like to see more installations expending effort to make the programming process easier and more enjoyable, since the shortfall in European software staff does not seem likely to disappear in the next few months.

We feel that it is only by consciously moving for more effective programming techniques that the user community will overcome the present difficulty with maintaining complex applications packages.

Software Product Description

Product Name: LOCK-11 Version 2.1 Description:

Lock-11 is a security superstructure built upon the standard RSTS password structure that provides the following extensions:

page 49

- Absolute control of system access by keyboard. Manager may limit any keyboard to certain accounts or groups of accounts and control time of access.
- Password knowledge is no longer carte blanche system access. System detects unauthorized use of passwords. Privileged passwords don't work on non-privileged keyboards. Nonprivileged passwords work only on specified keyboards.
- Real time system surveillance. Manager specifies a list of alarm keyboards which log all infractions and probes as they happen. Opser is not required.
- Auto-login (with or without password) and chain with specified core common contents by KB.
- Manager may establish special priority/burst settings by KB. Manager may establish default output protection code, @ assignment and up to three specific user logicals for each KB. All assignments are made at log-in.
- Manager specifies a list of console keyboards from which security file editor may operate.
- Manager may define a KB-specific access-denied message.
- A macro DYNPRI program is included which performs the following functions:
 - Users may be dispatched into ten separate priority queues, separately tunable on-line. Each queue has ten levels. Queues are selectable by KB.
 - Program detects hibernating jobs and announces the fact on ALARM keyboards. Privileged jobs hibernating cause extra loud and long alarms.
 - The program produces almost no load in operation and runs in 5K words.
 - Program will hold up to fourteen files open for performance purposes.

Minimum Hardware/Software Required:

Any valid RSTS/E system running Version 7.0 or later. Any version of RSX emulation is needed.

Support: See License Agreement

Installation: User Installed

Ordering Information:

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

First license	0% discount
Second thru Third license	50% discount
Fourth thru Twentieth license	70% discount

Licensed users desiring source code for internal use only must execute a separate Program Sources License Agreement. Sources are available at ten times the initial license fee.

License Fee:

Single CPU license: \$950.00. Annual maintenance at 12% of current list price.

Contact:

Dave Mallery Nationwide Data Dialog 215—364-2800

CIRCLE 12 ON READER CARD

\$REORDR — Sorting Alphabetically

By George T.A. May, Special Systems Division, Software Sciences Limited, UK

RSTS professionals may be interested in this slight variation to \$REORDR which allows it additionally to sort the directory in alphabetic order as well as the normal date and time sorts. This can be useful if you want to look for a particular filename in a directory. I have found it helpful for program development directories where one tends to have a mass of files and date order is not so interesting, but

INSTALLATION

Read	19
OLD	E1,200JREORDR.OLD

Ready

DELETE 4030,15210 Reads

REPLACE REORDR. TMP

Ready

OLD REORDR. TMP

Ready

APPEND REORDR. APP

Ready

COMPILE

Ready

FIF E1,1403/LI

DEMONSTRATION

Name .Ext	Size	Prot	Date	SY:E1,140]	
HUNDE ILNU					
COMMON.MAC	31	< 40>	30-Apr-80		
HKI010.HLP	2	< 60>	19-May-81		
DWAM .BAC	310	<124>	10-Jun-81		
ERMKBO.MAC	10	< 60>	12-Jun-81		
HKI010.OLD	760	<124>	05-Jun-81		
HKI010.LST	87	< 60>	16-Jun-81		
IV050 .BAK	156	< 60>	16-Jun-81		
UKBP02.DAT	163	< 60>	14-May-81		
IV050 .CBL	156	< 60>	17-Jun-81		
POL .CNV	- 2	< 60>	17-Jun-81		
IMV .CNV	2	< 60>	17-Jun-81		
HKI010.V01	32	< 48>	17-Jun-81		
STDO1 .OBJ	23		13-May-81		
STD01 .BAK	24		13-May-81		
STD01 .CBL	24	< 60>	17-Jun-81		
STD01 .SKL	1	< 60>	13-May-81		
PRO30 .OLD	81C	<124>	05-May-81		
PR030 .BAK	34	< 60>	05-May-81		
PR030 .CBL.	35	< 60>	17-Jun-81		
FOLL .CNV	2	< 60>	17-Jun-81		
INV1 .CNV	2	< 60>	17-Jun-81		
HKI010.BAK	33	< 48>	17-Jun-81		
HKI010.CBL	33		17-Jun-81		
TEMP49.TMP Total of 107	36 6 blocks		17-Jun-81 files in S)	7 (:[1,140]	
Total of 107					
Total of 107	6 blocks	in 24	files in S)		
Total of 1070 Ready RUN REORDR REORDR 97.00	6 blocks -08 RSTS	in 24 V7.0-(files in S) 07 HR-AUDIT		
Total of 1074 Ready RUN REORDR REORDR V7.00 Directory Red	6 blocks -08 RSTS ordering	in 24 V7.0-(Prosra	files in S) D7 HR-AUDIT BM		
Total of 107 Ready RUN REORDR REORDR V7.00 Directory Rei Sort Directo:	6 blocks -08 RSTS ordering ry(s) (Y	in 24 V7.0-(Prosr ES/NO)	files in S) D7 HR-AUDIT Sm <nd>? Y</nd>	r:[1,140]	
Total of 107 Ready RUN REORDR REORDR U7.07 Directory Rei Sort Director Order by CREi	−08 RSTS ordering ry(s) (Y ation Da	U7.0-(Prosra ES/NO) te/Time	files in S) D7 HR-AUDIT am <nd>? Y c, ACCess Da</nd>		
Total of 1074 Ready RUN REORDR REORDR V7.00 Directory Rei Sort Directo	−08 RSTS ordering ry(s) (Y ation Da	U7.0-(Prosra ES/NO) te/Time	files in S) D7 HR-AUDIT am <nd>? Y c, ACCess Da</nd>	r:[1,140]	
Total of 1074 Ready RUN REORDR REORDR V7.00 Directory Red Sort Director Order by CRE In FORIWARDJ Device and UF	-08 RSTS ordering ry(s) (Y ation Da or REVE FD Speci	in 24 V7.0-(Prosr: ES/NO) te/Time ERSE] (ficatio	files in S) D7 HR-AUDIT am <nd>? Y e, ACCess Da Drder<for>? on(s)? DRO1</for></nd>	<pre>tte, or ALPhabetically <cre>? ALP FOR E1,140]</cre></pre>	
Total of 107 Ready RUN REORDR REORDR 07.00 Directory Red Sort Director Order by CREJ In FORIWARDJ Device and Uf Directory _DF	6 blocks -08 RSTS ordering ry(s) (Y ation Da or REVE FD Speci RO:E1.14	in 24 V7.0-(Prosr: ES/NO) te/Time ERSEJ (ficatio 0] has	files in S) D7 HR-AUDIT Sm <nd>? Y P, ACCess Da Drder<for>? Drder<for>? Drder<for>? Drder<for? Drder<for??< td=""><td><pre>tte, or ALPhabetically <cre>? ALP FOR E1,140]</cre></pre></td><td></td></for??<></for? </for></for></for></nd>	<pre>tte, or ALPhabetically <cre>? ALP FOR E1,140]</cre></pre>	
Total of 107 Ready RUN REORDR REORDR V7.00 Directory Red Sort Director Order by CREJ In FOREWARDJ Device and Uf Directory _D	6 blocks -08 RSTS ordering ry(s) (Y ation Da or REVE FD Speci RO:E1.14	in 24 V7.0-(Prosr: ES/NO) te/Time ERSEJ (ficatio 0] has	files in S) D7 HR-AUDIT Sm <nd>? Y P, ACCess Da Drder<for>? Drder<for>? Drder<for>? Drder<for? Drder<for??< td=""><td><pre>tte, or ALPhabetically <cre>? ALP FOR E1,140]</cre></pre></td><td></td></for??<></for? </for></for></for></nd>	<pre>tte, or ALPhabetically <cre>? ALP FOR E1,140]</cre></pre>	
Total of 1074 Ready RUN REORDR REORDR V7.0- Directory Red Sort Director Order by CREd In FOREWARDJ Device and UF Directory _DF Sort Director	6 blocks -08 RSTS ordering ry(s) (Y ation Da or REVE FD Speci RO:E1.14	in 24 V7.0-(Prosr: ES/NO) te/Time ERSEJ (ficatio 0] has	files in S) D7 HR-AUDIT Sm <nd>? Y P, ACCess Da Drder<for>? Drder<for>? Drder<for>? Drder<for? Drder corde</for? </for></for></for></nd>	<pre>tte, or ALPhabetically <cre>? ALP FOR E1,140]</cre></pre>	
Total of 107 Ready RUN REORDR REORDR V7.0- Directory Rei Sort Director Order by CRE: In FOREWARDJ Device and UF Directory _DF Sort Director Ready	6 blocks -08 RSTS ordering ry(s) (Y ation Da or REVE FD Speci R0:[1:14 ry(s) (Y	in 24 V7.0-(Prosr: ES/NO) te/Time ERSEJ (ficatio 0] has	files in S) D7 HR-AUDIT Sm <nd>? Y P, ACCess Da Drder<for>? Drder<for>? Drder<for>? Drder<for? Drder corde</for? </for></for></for></nd>	<pre>tte, or ALPhabetically <cre>? ALP FOR E1,140]</cre></pre>	
Total of 107 Ready RUN REORDR REORDR 07.0- Directory Rei Sort Director Order by CRE: In FORIWARDJ Device and Uf Directory _DF Sort Director Ready PIP E1+1403/1	6 blocks -08 RSTS ordering ry(s) (Y ation Da or REVE FD Speci Arg(s) (Y LI	in 24 V7.0-(Prost ES/NO) te/Time ERSE] (ficatio 0] has ES/NO)	files in S) D7 HR-AUDIT m <nd>? Y e, ACCess Da Order<for>? on(s)? DRO: been Reorde <nd>? ~C</nd></for></nd>	<pre>tte, or ALPhabetically <cre>? ALP FOR [1,140] red</cre></pre>	
Total of 1074 Ready RUN REORDR REORDR V7.0- Directory Rei Sort Director Order by CRE: In FOREWARDJ Device and UF Directory _DF Sort Director Ready PIP E1+140J/H Name .Ext	<pre>6 blocks -08 RSTS ordering ry(s) (Y ation Da or REV[FD Speci R0:[1,14 ry(s) (Y</pre>	in 24 V7.0-(Prosr; ES/NO) te/Time ERSE] (ficatio 0] has ES/NO) Prot	files in SY D7 HR-AUDIT Sm <nd>? Y e, ACCess Da Drder<for>? On(s)? DRO! been Reorde <nd>? °C Date</nd></for></nd>	<pre>tte, or ALPhabetically <cre>? ALP FOR E1,140]</cre></pre>	
Total of 1074 Ready RUN REORDR REORDR V7.00 Directory Red Sort Director Under by CRE4 In FORIWARDJ Device and UF Directory _DF Sort Director Ready PIP E1+140J/H Name .Ext COMMON.MAC	<pre>6 blocks -08 RSTS orderind ry(s) (Y ation Da or REVE ED Speci R0:E1.14 ry(s) (Y LI Size 31</pre>	in 24 V7.0-(Prosri ES/NO) te/Time ERSEJ (ficatio 0] has ES/NO) Prot < 40>	files in S) D7 HR-AUDIT AND>? Y P, ACCess Da Drder <for>? on(s)? DRO! been Reorde <nd>? "C Date 30-Apr-80</nd></for>	<pre>tte, or ALPhabetically <cre>? ALP FOR [1,140] red</cre></pre>	
Total of 1074 Ready RUN REORDR REORDR V7.0- Directory Rei Sort Director Order by CRE: In FOREWARDJ Device and UF Directory _DF Sort Director Ready PIP E1+140J/H Name .Ext	<pre>6 blocks -08 RSTS ordering ry(s) (Y ation Da or REV[FD Speci R0:[1,14 ry(s) (Y</pre>	in 24 V7.0-(Prost ES/NO) te/Time ERSEJ (ficatio Ol has ES/NO) Prot < 40> <124>	files in SY D7 HR-AUDIT Sm <nd>? Y e, ACCess Da Drder<for>? On(s)? DRO! been Reorde <nd>? °C Date</nd></for></nd>	<pre>tte, or ALPhabetically <cre>? ALP FOR [1,140] red</cre></pre>	

names are. The following source file \$REORDR.BAC < 124 > or whatever it is called in a particular system.

I've attached an installation process and demonstration run. You've got to follow the installation process else RSTS Basic Plus routine system screws you up because I've redefined function FNI%.

HKI010	.BAK	33	< 48>	17-Jun-81
HKI010	O.CBL	33	< 48>	17-Jun-81
HKI010	HLP.	2	< 60>	19-May-81
HKI010	0.LST	87	< 60>	16-Jun-81
HKI010	D.OLD	760	<124>	05-Jun-81
HKI010	0.001	32	< 48>	17-Jun-81
INU	.CNV	2	< 60>	17-Jun-81
INV1	. CNV	2	< 60>	17-Jun-81
IV050	.BAK	156	< 60>	16-Jun-81
IV050	+CBL	156	< 60>	17-Jun-81
POL	.CNV	2	< 60>	17-Jun-81
POLL	.CNV	2	< 60>	17-Jun-81
PR030	, BAK	34	< 60>	05-May-81
PR030	+ CBL	35	< 60>	17-Jun-81
PR030	.OLD	810	<124>	05-May-81
STD01	.BAK	24	< 60>	13-May-81
STD01	.CBL	24	< 60>	17-Jun-81
STD01	.OBJ	23	< 40>	13-May-81
STD01	.SKL	1	< 60>	13-May-81
TEMP 49	7.TMP	36	< 60>	17-Jun-81
UKBP02	2.DAT	163	< 60>	14-May-81

Total of 1076 blocks in 24 files in SY:[1,140]

Ready BYEF

89

12

21	EXTEND 1 VER/ED	EDIT DATE	REASON	5	
	1 7.0-08	15-JUN-81	ORDER ALPABETICALLY	*	
410	1 AX 1 AUX 1 BX 1 CX	POINTER TO FI	EXT AVAILABLE TARGET UFD ENT IRST BLOCK BEYOND LAST NAME UCKS IN UFD & IZE &		
	CS C3x	USER ENTERED	INPUT & ER BY ACCESS DATE &		
	0.5	= 1 IF SORTI	ER BY CREATION DATE/TIME & NG ALPHABETICALLY & CIFICATION OF UFD CURRENTLY	BEING &	
	03	PROCESSED &	LE FOR INTERNAL FUNCTIONS &	521-16 G	
	1 01()	THE UFD TO BE	E REORDERED SET UP AS A WORD		
	FX() FOX	AORKING VARIA	DUNTER FOR EACH DIRECTORY HL	DCK &	
	1 F1X	WORKING VARIA	ABLE & #'S TEXT STRING &		
	i Iz()	ARRAY USED TO	HOLD RETURN FROM FILE STRI	NG SCAN &	
00	1 SX(.)		RED DEVICE/UFD SPECIFICATION RTING NAMES BY ACCESS OF CRE		
	5%	COUNT OF NUME RETRIEVAL ENT	BER OF ATTRIBUTE, ACCOUNTING TRIES FOR CURRENT FILE BEING	AND &	
	50%	= 1% IF SC	DRT BY ACCESS DATA DESIRED &		
	SP1%		PARAMETER TO FNIX - REPLACE		
	1 SP21 SP31	= SECONDSORT = THIRD SORT	PARAMETER TO FNIX - REPLACE PARAMETER TO FNIX - NEM	STX 8	
	1 11	WORKING VARIA	ABLE &		
			ET UP AS A NURD ARRAY &		
	1 T1X	POINTER TO NA	ABLE USED IN MOVING UPD CLUS AME ENTRY. USED IN CREATION/	ACCESS SORT &	
	1 121	POINTER TO AC SORT. &	COUNTING ENTRY, USED IN CRE	ATION/ACCESS &	
	l ws	FILE SPEC FOR	R UFD TARGET ARRAY. &		
	wix	WORKING VARIA			
	2%	NORKING VARIA			
52	I FNSAAPX()		WHEN SURTING ALPHANETICALLY		
	1		A FUNNY ORDER FOR THE NUMBER NUMBERS DECUR FOR HIGH (NOT	RS (IE =VE &	
	:		VALUES. FNSAAPX REUNDERS TH	EM PROPERLY &	
	1		BY ADDING 32768 TO -VE VALU -ING 32768 FROM +VE VALUES		
1	1		-	•	
99	I FNIX(LX,DX		GIVEN DIRECTORY LINK, DATE NAME ENTRY SORT LIST ARRA	Y IN SORTED URDER	8
	FNO%()	16000	CHECK TO SEE IF THIS IS A L AND IF SO, IF ANY FILES ARE ACCOUNT TO BE REURDERED.	ARGE FILE SYSTEM & OPEN ON THE &	
010	15 = "V7.0-08" SET UP VERSION/	EDIT #'S FOR H	EADER	*	
205	PRINT INPUT #31, "ORDI	ER BY CREATION	DATE/TIME, ACCESS DATE, OR	ALPHANETTCALLY CO	PF57 "
	GET USER SELECT			8	
210	GOTO 1210 IF LE	V(C\$) = 0%			
	CS = LEFT(CVTSS			6	
	GOTO 1220 IF CS GOTO 1205 IF CS	<> "ACC" AND	C\$ <> "ALP"	1	
	C31 = +11 IF C3			6	
	C3% = 1% IF C5 : PRESET TO DEFAUL	T VALUE - DOW	E IF DEFAULT INPUT. ELSE,		
	GET FIRST THREE	CHARACTERS IN	UPPER-CASE ONLY. DONE IF	:	
	"CRE". GET AG	IN IF NOT "AC	C" OR "ALP". OTHERWISE.	*	
005	PX = UX SX(0X,0X) = 0X				
	NULENTE = OE			6	
	NULENTX = -3276	TX-1X IF C3X =	11	6	

;	SX(0X,2X) = NULENTX 6 SX(0X,2X) = NULENTX 6	1 15210	DEF* FNIX(LX,SP1X,SP2X,SP3X) &
1	INITIALIZE POINTER INTO SOURCE ARRAY. ZERO NUMBER OF & ENTRIES IN FILE. IF SORTING BY DATE AND TIME SET LOWEST KEYS TO &	1	IF C31=11 THEN
	ZERU. IF SURTING ALPHABETICALLY, SET TO LANGEST -VE NUMBER TO	15225	SP11=FNSWAP2(SP11)
	CATER FOR -VE RADSO VALUES. NOTE TIMES WILL BE NEVERSED TO GIVE &	1	SP21=FNSHAP1(SP21)
	MINUTES BEFORE MIDNIGHT SO THAT THERE ARE NO 'SPECIAL CASE'	`	SP3%=FNSWAP%(SP3%)
	SORTS		IF DOING AN ALPHABETIC SORT, RADSO PRODUCES AVE NUMBERS
030	IF C3% = 0% THEN &	1	IN & FUNNY PLACE. SO REDROER THE NUMBERS. 6
	ZX=FNIX(LX,DX(T2X+3X),1440X=DX(T2X+4X),0X) 6	1	
	GOTO 4035 6	15230	GOTO 15370 IF SOX = 0X 8 GOTO 15370 IF SP1X > SX(0X,1X) 8
	SORTING BY CREATION DATE AND TIME 6		GOTO 15370 IF SP1% = S%(0%,1%) &
	NOT TIMES ARE SWAPPED FROM ORIGINAL VERSION TO MINUTES AFTER &		ANU SP2% > S%(0%,2%) &
	MIONIGHT	`	GOTO 15370 IF SP1% = S%(0%,1%) & & & & & & & & & & & & & & & & & & &
1031	IF C31 = -11 THEN		AND SP31 > S1(01,31) 8
	Zx=FNIX(LX,0X(12X+1X),0X,0X) &	1	SKIP SEARCH AND COMPARE IF NEW ENTRY GUES AT END OF LIST &
4	GOTO 4035 8	1	FOR W11 = 31 TO S1(01,01) + 21
	SORTING BY DATE OF LAST ACCESS	15330	FUR 114 - 34 10 34(U4)U4) + CA
40.52	IF C3x = 1X THEN 6		
	2x=FNIx(Lx, Dx(Px+1x), Dx(Px+2x), Dx(Px+3x))		GOTO 15380 IF SP1% < S%(x1%,1%) 8 GOTO 15380 IF SP1% = S%(x1%,1%) 8
	SORTING ALPHABETICALLY ON FILENAME (WORDS 1 AND 2) AND .EXT &		AND SP2% < S%(A1%,2%)
1	OBTAINED FROM THE UED NAME BLOCK	· ·	GOTO 15380 IF SP1% = S%(#1%,1%) 8
1			AND SP2X = SX(A)X,2X) AND SP3X < SX(A)X,3X) 8
152011	FUNCTION FNIX(LX, SPIX, SP2X, SP3X)	`	NEXT A12 8
1	6 C	1	FOR ALL ENTRIES IN THE LIST, SEARCH FOR THE FIRST ENTRY &
1	THIS FUNCTION INSERTS RECORDS, IN ASCENDING 'KEY' SEQUENCE, &		ABOVE THE NEW ENTRY TO BE MADE &
	INTO AN IN-CURE ARRAY. 6	15370	M11 = S1(01,01) + 31
:	THE IN-CORE ARRAY CONSISTS OF 1738 3-WORD RECORDS. THE FIRST &		SX(0X,1X) = SP1X 6
1	THREE RECORDS ARE USED FOR HOUSEKEEPING PURPOSES, WHILE THE		SX(0X,2X) = SP2X 6
1	LAST 1735 MAY CONTAIN VALID DATA.	1	SX(01,31) = SP31 8
1	EACH VALID RECORD CONSISTS OF A RSTS/E LINKWURD(THE ADDRESS) 6	i	SET LUOP FOR SINGLE ENTRY ADD AT END OF LIST. SAVE
1	AND THWEEE INTEGERS (DATE,0,0/DATE,TIME,0/FILE,NAME,EXT.) & DEPENDING UN THE SORT TYPE SELECTED BY THE USER, 1ST RECORD &	-	KEY AS HIGHEST IN FILE
1	(0.0) IN THE ANRAY CONTAINS THE CURRENT COUNT OF VALID RECORDS, &	15380	SX(21,01) = L1 8
:	AND THE SONT PARAMETERS OF THE HIGHEST RECORD ENTERED. NEXT TWO &	1	Sz(22,12) = SP12 &
4	RECORDS ARE USED FOR TEMPORARY STORAGE DURING THE EVTRY PROCESS.	×	SX(2X,2X) = SP2X &
1	FNIX IS CALLED WITH THE FULLOWING ARGUMENTS:		Sx(2x, 3x) = SP3x Sx(0x, 0x) = Sx(0x, 0x) + 1x
1	6	i	SAVE ENTRY FOR LOADING INTO LIST, INCREMENT TOTAL NUMBER 6
1	LX - LINKHURD TO SOURCE UFD NAME ENTRY BLOCK 6	1	UF ENTRIES IN LIST 6
1	SP1% - EITHER DATE OF CREATION, OR DATE OF LAST & ACCESS ON FIRST FILE NAME WORD FOR NAME BLOCK &	15 490	FOR WX = W1X TO SX(0x,0X) + 2%
1	POINTED TO BY LX	15510	FUN T11 = 01 TO 51 8
1	SP2X - TIME UF CREATION OR OX IF SURTING BY ACCESS DATE &	1	SET OUT AND INNEH LOUP CONTHULS FOR DATA MOVEMENT &
1	ON SECOND FILENAME WORD & SP3% - 0% FOR DATE SORIS OR FILENAME EXTENSION &	17000	DEF* FNSWAPX(#1%)
-	SPST - OT FOR DATE SURIS OR FILENARE EXTENSION	17000	A00 = 32768. 6
1	ALL FILENAMES AND EXTENSIONS ARE HELD IN HAUIX 50 6	í,	ADD = -AUD IF WTX >= 0%
1	A THAN A THEN TO THEFT & HEADER HAVE A THAN A	1	AT = WIX + ADD & S
1	IF THE CALLER ATTEMPTS TO INSERT A RECORD WHOSE KEY IS > THAN & THE HIGHEST KEY IN THE FILE, THEN THE SEARCH AND COMPARE PORTIONS	1	FNEND 6
1	OF THE ALGORITHNM IS SKIPPED.	1	RADIA SO STRINGS DO NOT APPEAR IN A MONOTONICALLY INCREASING &
1	6	1	ORDER. THIS FUNCTION RE-ORDERS THEM SO THAT THEY DU. 6
-	THE FUNCTION RETURNS ON IF INSERTION OCCURRED, AND -1%(TRUE) & IF INSERTION DID NOT OCCUR.	\$2767	END
:	6	18131	· · · · · · · · · · · · · · · · · · ·
1	IF SUX IS RESET (02), NO SORT IS DUNE AND THE INSERTION BECOMES &		
1	A SIMPLE STRAIGHT LINE INSERTION. 6		

RSTS/E SOFTWARE PACKAGES

- KDSS, a multi-terminal key-to-disk data entry system. (Also available for RSX-11M.)
- TAM, a multi-terminal screen-handling facility for transaction-processing applications. (Also available for RSX-11M.)
- **FSORT3**, a very fast sort. Directly sorts RSTS/E files containing up to 16 million keys or records. Up to 70 times as fast as the RSTS-11 Sort package in CPU time.
- SELECT, a convenient, very quick package for extracting records that meet user-specified selection criteria.
- **BSC/DV**, a device driver for the DEC DV11 synchronous multiplexer that handles most bisynchronous protocols.

- COLINK, a package that links two RSTS/E systems together using DMC11s. Supports file transfers, virtual terminals, and across-thelink task communication.
- DIALUP, a package that uses an asynchronous terminal line to link a local RSTS/E system to a remote computer system. Supports file transfers, virtual terminals, and dial-out through a DN11.

(The performance-critical portions of the first five packages are implemented in assembly language for efficiency.)

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

CIRCLE 29 ON READER CARD

TIPS & TECHNIQUES

A Column For The Advanced RSTS/E User

In this issue, we will look at a patch to add form names to the one-shot spooling sys-call, the time diferences between Basic-Plus and Basic-Plus-2 (or CSPCOM) CUSPs (Commonly Used System Programs), and the memory differences between Basic-Plus and Basic-Plus-2 CUSPs.

Correction!

In the last issue, I presented a patch to set a VT100 to VT52 mode upon exiting EDT V2. The patch functions by printing the escape sequence to change to VT52 mode. Due to the printing font, the last character in that escape sequence looks like a "1" (ASCII 49), when it should be a lower case "I" (ASCII 108).

One-Shot Spooling Sys-Call Form names

The release of RSTS/E V7.0 added a new sys-call called the one-shot spooling sys-call. This call allows a user program to submit requests to the spooling system (SPOOL and BATCH). Now, programs can be written that interface to the spooling system directly, instead of chaining back and forth to QUE.

Unfortunately, the implementation of the call requires all parameters to be passed in FIRQB, instead of using XRB to point to the command string (like the .FSS call). This means that there is not enough room to specify a form name. This is a limitation to systems that either print on a variety of forms, or use the form name to allow the operator to select which request gets printed when.

There is not enough free room in the call to allow the specification of the form name, but there is a free bit in the flag word passed in the call. This patch tells QUMRUN that if the 6'th bit (value = 64) is on in the flag word to use the first six characters of the file name as the form name.

```
RUN [1,2]CPATCH
CPATCH V7.0-07 RSTS V7.0-07 Softec Dev 11/70
File to patch - QUMRUN.BAS = QUMRUN.BAS
#KB:/CS:1299
*H/2!/V<cr>
2!<tab><tab>PROGRAM<tab><tab>:QUMRUN.BAS<cr>
*H/10140/V<cr>
10140<tab>M$=JOBP$ &<cr>
*OAI<cr>
10135 < tab>IF < tab>ONE.SHOT% & < cr>
<tab>THEN<tab>IF<tab>(ASCII(MID(M$, 4%, 1%)) AND 64%) &<cr>
<tab><tab>THEN<tab>LSET FORM$ = MID(M$, 9%, 4%) &<cr>
<tab><tab>! IF THIS IS A ONE-SHOT SPOOLING CALL, &<cr>
<tab><tab>! THEN IF THEY SET OUR FORM NAME BIT IN THE FLAG WORD, &< cr>
<tab><tab>! THEN JAM IN THE FILE NAME AS THE FORM NAME. &<cr>
< CT>
<esc>*V<cr>
10140 < tab > M$ = JOBP$ & < cr >
*EX<cr>
Patch from -KB:[x,x].CMD complete.
#17
File to patch - 1Z
```

By Steven L. Edwards, Software Techniques

After installing this patch, if you set the 6'th bit in the flag word (as in the example below) when you send a spooling request, the file name will be used as the form name.

```
    3010 TEMP.0$ = SYS(CHR$(6) + CHR$(-28) + CVT%$(0) &
+ MID(SYS(CHR$(6) + CHR$(-10) + FILE.NAME$), 5, 8) &
+ STRING$(6,0) + CVT%$(SWAP%(4 + 64))) &
    ! QUE THE FILE TO "LP:" USING THE FILE NAME AS THE FORM NAME, &
! AND FLAG IT TO BE DELETED WHEN DONE. &
```

Time differences between Basic-Plus and Basic-Plus-2 CUSPs

My initial goal was to demonstrate the time (clock and cpu) differences between Basic-Plus and Basic-Plus-2. I figured that the demonstration must be directed towards some pratical end, so I chose 4 of the CUSPs that I felt were particularly 'piggy.' The CUSPs I chose were ANALYS + ANALY1, BACDIR, DIRECT, and REORDR. This would show us the benefits to be gained just by using the Basic-Plus-2 compiler instead of the Basic-Plus interpreter.

This was not particularly fair to Basic-Plus-2 because it was not designed to be a faster Basic-Plus. What is efficent in one language is not always efficent in another language.

So, I decided to make some minor modifications to the programs to make them more efficent for Basic-Plus 2. Since all of these programs use virtual arrays, and Basic-Plus-2 will do 'program internal' random caching of virtual array file blocks if given the memory, we open the files with a recordsize that is a multiple of 512+6 (6 bytes to keep track of which block is where). I chose a recordsize of 4144 ((512+6)*8) which meant that we were reducing the potential disk activity by a factor of 8 in exchange for a little more than 2 KW of memory.

The experiment was as follows:

- Run each program in Basic-Plus (BAC), Basic-Plus-2 (TSK), and Basic-Plus-2 with large recordsizes (LRS) 12 times, recording the clock and cpu time consumed in each run.
 - 1. Run ANALYS, telling it to output to NL:.
 - Replace BACMNT with a program that just exits, run BACKUP, telling it to backup one large account. Note that this means that the times presented below include the time spent on BACKUP and BACCON.
 - 3. Run DIRECT, telling it to output a /S listing of one large account to NL:.
 - 4. Run REORDR, telling it to sort one large account.
- 2. Discard the lowest, and the highest times to reduce the chance of a stray value influencing the result.
- 3. Calculate the mean times.
- The results of the experiment are:

"I (for once) was speechless." - Dave Mallery, March, 1981 issue of RSTS PROFESSIONAL

To tell the truth, so are we. We knew DISKIT would amaze RSTS users, but, frankly, we were unprepared for the response. Phone calls, letters, and now the RSTS PROFESSIONAL -- all saying what we want you to know:

DISKIT is a remarkable

software tool!

Listen to what else Dave has to say:

"...using DISKIT, I created 130 accounts and fully extended their centered UFDs in 3 minutes and 40 seconds (a job that used to take 4 to 8 hours.)"

"I then copied the full contents of a 300 MB RM05 equivalent to this new 'well-structured' disk in 45 minutes, optimizing clustersize and contiguity in the process..."

DISKIT IS A DISK STRUCTURING UTILITY

As Dave discovered, DISKIT's disk structuring utility, DSU, is fast. It also:

- Optimizes file clustersizes
- Places and pre-extends UFDs
- Performs transfers between unlike disks
- Saves all accounting data
- Allows manual file placement
- Provides full logging and statistics
- Includes sophisticated error handling and recovery

DISKIT IS A DIRECTORY PROGRAM

But DISKIT is more than a disk structuring utility. DISKIT's Macro-11 directory program, DIR, displays directories 12 times faster than before -- looking up files by name, extension, and date (with wildcards) at the incredible rate of 250 files/second.

And DIR is smart. It supports all standard DIRECT switches (including backwards, up to 1,000 files) with features you won't find elsewhere -like password lookup, UFD placement, and UFD size.

DIR even works as a diagnostic tool on dismounted disks, detecting bad directory structures and identifying them with comprehensive error messages.

DISKIT IS AN OPEN FILES DISPLAY PROGRAM

DISKIT's Macro-11 OPEN program displays open files by job – with complete job and file statistics. It even has a "sleep switch", allowing you to dynamically update information at any desired interval.

DISKIT LETS YOU WRITE YOUR OWN DISK HANDLING ROUTINES

Best of all, the very same routines used in DISKIT are included, with documentation, so you can write your own disk handling routines. In minutes.

DISKIT IS THE FIRST SOFTWARE TOOL KIT FOR COMPLETE DISK MANAGEMENT

The DISKIT package provides all the tools and utilities you need to create and manage a wellstructured disk. The entire DISKIT package, with extensive documentation is available now for only \$1250.

DISKIT, Dave says, "...is the 'final solution' to structured disks, eliminating all of the time and complexity and reducing the job to one of a SAVRES."

What more could we say?

Once again, we've got the answer.

Software Techniques, Inc.

5242 Katella Avenue, Los Alamitos, CA 90720 [714] 995-0533 [213] 594-9405 CIRCLE 65 DN READER CARD

RSTS DEFECTOR

By Joel Schwartz, M.D.

I have a confession to make. I have defected from RSTS. But before you put my article down let me explain. I've played all the games available to me on the RSTS system so I went in search of new game horizons on other computers and here's what I found. There seems to be two basic types of games available. One is the Arcade type game. Space Invaders and Astroids would be examples of this type. The other is the Adventure D and D type of which Adventure and Dungeon would be an example. The games can be found on 51/4 floppy discs and can be run on computers that are named after a fruit, a household animal, a naval officer and one that begins and ends with a vowel. (I had to disguise these in order to keep my job.) If you have any trouble figuring them out, send a self-addressed envelope to me and I will send the answers.

I want to tell you about two of the latter type games. The first is called ULTIMA (California Pacific Computer Co. by Lord British). Billed as the ultimate adventure game, in

Program	Clock	%/BAC	Сри	%/BAC	KW(prgrm + shared)
ANALYS.BAC	333.5	100.0	301.3	100.0	32(16 + 16)
BACDIR.BAC	103.0	100.0	14.4	100.0	31(15 + 16)
DIRECT.BAC	40.9	100.0	30.1	100.0	30(14 + 16)
REORDR.BAC	176.7	100.0	95.7	100.0	30(14 + 16)
Using the Basic-I	Plus run-tim	ne system			
ANALYS.TSK	196.9	59.0	161.4	53.6	31(15 + 16)
BACDIR.TSK	121.9	118.4	10.0	69.4	30(14 + 16)
DIRECT.TSK	45.1	110.3	20.4	67.8	29(13+16)
REORDR.TSK	122.0	69.0	33.8	35.3	29(13+16)
Using the Basic-I	Plus-2 run-t	ime system			
ANALYS.LRS	180.4	54.1	160.9	53.4	29(21 + 8)
BACDIR.LRS	24.0	23.3	6.6	45.8	30(22 + 8)
DIRECT.LRS	24.1	58.9	20.5	68.1	27(19+8)
REORDR.LRS	44.2	25.0	26.0	27.2	27(19+8)
Using the Basics	resident lib	orary			

From these results we can see that a program written to be efficient in Basic-Plus may run faster or slower in Basic-Plus-2, but with a little effort, programs can run considerably faster. For our system, this minor modification reduced the clock time for BACDIR to get a directory of the whole system from over 12 minutes to just over 1 minute. This made running BACKUP during timesharing almost painless.

Based on the results of this experiment, I plan to modify all of the programs in the spooling and backup packages to use large recordsizes on virtual array file opens.

Multi-user Tasks

Multi-user tasks are tasks that have been task-built to allow the separation of read-only code from read-write code. For RSTS this means separating the read-only code into a separate resident library. This means that when a user runs a program, RSTS will load the resident library and the readwrite section of the task. When a second user runs the same program, RSTS will just load the read-write section of the task. The exact procedures to create a multi-user task have been published several times in several publications, and will not be repeated here. (See Vol 3-2 p 8, June 1981.)

As a practical example of how multi-user tasks can reduce total system memory requirements, we will look at a

many ways that's exactly what this game is. There is the ability to create new characters of different races (human, elf, hobbit, dwarf) and types (fighter, cleric, wizard, thief). Once formed the character can travel on land, across oceans, in the air, and eventually into space. Towns, castles, islands and dungeons are but a few of the places on the adventurers way. The game is filled with quests, challenges and monsters which test your skill and creativity. A well documented manual and separate player reference card come with each game. I have been deliberately vague, for to be more specific would only spoil the game.

The second game I enjoyed was THE PRISONER (Edu-Ware, Conoga Park, California). Based on the popular English TV program, the object of the game is to escape from the ISLAND where you are being held prisoner. While you are trying to do this the computer is trying to get you to reveal three digit secret code number it gives you at the beginning of the game and forewarns you never to reveal. Every psychological trick in the book is used to obtain your information. If you are easily frustrated, this is not the game for you. However, if you are ... but that would be telling.

real-life example:

One of our clients is a heavy user of the spooling system. During the morning operator's shift, 5 print spoolers run almost constantly. All CUSPs are running in Basic-Plus-2, linked with the Basics resident library.

SPLRUN runs in 18KW, so if we multiply these numbers out we get 90KW of print spoolers competing for memory. If we build SPLRUN as a multi-user task, we get a SPLRUN that runs in 3KW, with a 15KW SPLRUN resident library. Since all 5 spoolers will share their 15KW read-only segment, they only require 30KW to run all 5 spoolers at the same time. This saves 60KW of their total system memory requirements by re-task-building a single program!

The only 'fly-in-the-ointment' is that resident libraries must be loaded at specific addresses. This is a serious limitation to systems with small amounts of physical memory. I hope DEC changes this in future releases of RSTS.

Quick Tip

On occasion, programmers need to execute a Basic[2] program starting at a non-zero line number. The usual way to accomplish this is to switch to the Basic-Plus run-time system and execute an immediate mode CHAIN specifying a non-zero line number. If you are in a run-time system other than Basic-Plus, type 'RUN programname/PO:linenumber'. If you are in the Basic-Plus run-time system type 'RUN programname < LF > linenumber.'

The reason this works is that when you execute a .RUN monitor call (which Basic[2] considers a CHAIN), you can specify a parameter word (@ FIRQB + FQNENT) which Basic[2] will interpret as the line number to start execution at. This is the same location in FIRQB that receives the POsition switch value after a file name string scan (.FSS).

If you have any questions or suggestions for things you would like to see in this column, please address them to:

Steven Edwards, Software Techniques, 5242 Katella, Los Alamitos, CA 90720

.

HARDGODU

Finally! A publication exclusively for DEC* or **DEC-related users, manufacturers, and consultants!**

What is HARDCOPY?

HARDCOPY is a bi-monthly magazine written and edited for the community of users and vendors of DEC and DEC-compatible equipment. Topics ranging from new peripherals to software updates give the reader a broader, more in-depth coverage than most all other publications.

Who reads HARDCOPY?

HARDCOPY is exclusively directed at the DEC marketplace. Along with its primary readership of end users, systems analysts, site managers, MIS directors, and engineers HARDCOPY also circulates to manufacturer consultants and OEM's of DEC-compatible peripherals and software vendors.

Inside HARDCOPY ... a magazine covering it all!

Five HARDCOPY sections - DEC NEWS, EDITORIAL, DEC COMPATIBLES. BUSINESS SPOTLIGHT, SOFTWARE and VAX INFO highlight all the latest topics of interest concerning small, medium and large scale computers.

Why should you subscribe?

If you are a user, manufacturer, consultant or OEM of DEC related equipment, HARDCOPY will provide you with highly informative, up-to-date news ranging from general interest articles to new product announcements.

To receive HARDCOPY M	lagazine fill out the	qualification form below
-----------------------	-----------------------	--------------------------

	IMPORTANT — To qualify, please answer all questions below. Type of DEC computer at this location:		
Subscription Form	CPU		
Name Title	Disk		
Company	Floppy		
Company Address	CRT	Other	
City/State/Zip	Primary product or service	of your firm at this location:	
Phone ()	End User	OEM (type)	
Signature	System House		
Date		Educator	
Check here if you wish to receive advertising information from Hardcopy.	Data Processing	Government	
Hardcopy.	Software	Other	
HARDCOPU is published six times per year by Seldin Pu	blishing, Inc. One year subs	cription may be purchased at the	
following rates:			
United States		\$16.00	
All Other Countries		\$25.00	
	CALIFORNIA 92621	Phone (714) 632-6924	



TTETERS PORTO

S-

user 11

^{\$}14,500 THROUGH JANUARY 31, 1982

APPLICATIONS SCALE IN YOUR FAVOR...WITH USER-11.

USER-11 is a comprehensive applications development facility for the DEC RSTS operating environment. Dozens of integrated programs harness RSTS's power for unparalleled productivity and performance in constructing on-line and batch application systems.

PRODUCTIVITY... A MATTER OF TIME.

More than a data management system, USER-11 features common-function programs that permit numerous applications to be installed without writing a single line of code. Complete building blocks and interfaces are provided for those remaining applications requiring custom work.

PERFORMANCE... SIMPLY INCREDIBLE.

USER-11 combines advanced BASIC and MACRO coding techniques with ultra-efficient file accessing mechanisms to optimize application system performance.

RELIABILITY... A PROVEN FACT.

USER-11 is currently installed on hundreds of timesharing systems world-wide with a reliability record that users repeatedly praise. All software is exhaustively tested and benchmarked prior to any distribution release.

SECURITY... MORE THAN RSTS.

USER-11 incorporates a unique MENU system which flexibly and securely controls all processes. Secondary, encoded security databases are provided for each project. A special Run Time System is invoked to prevent accessing the RSTS ready state, unless the software developer desires this for the user.

STANDARDIZATION... A BYPRODUCT.

All USER-11 generated packages employ programming and documentation conventions which enhance compatibility, readability, and maintainability.

ADAPTABILITY... NO PROBLEM.

USER-11 programs are dictionary and parameter driven throughout. Files can be restructured without program modifications.

DOCUMENTATION... GOOD AND PLENTIFUL.

USER-11 features a wealth of easy-to-follow documentation. An extensive on-line "/HELP" facility is at software developer and user fingertips. All documentation is maintained and distributed on your system's compatible media.

TRAINING...ALL KINDS.

USER-11 training courses are held frequently with instructional programs to suit your need—beginner to expert.

FEATURES...ON AND ON.

USER-11 includes virtually every facility needed to quickly construct high performance management applications—nothing else is required. If you find this hard to believe or would like more information, contact us; we will furnish you with solid user proof!



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

DEC and RSTS are registered trademarks of Digital Equipment Corporation.

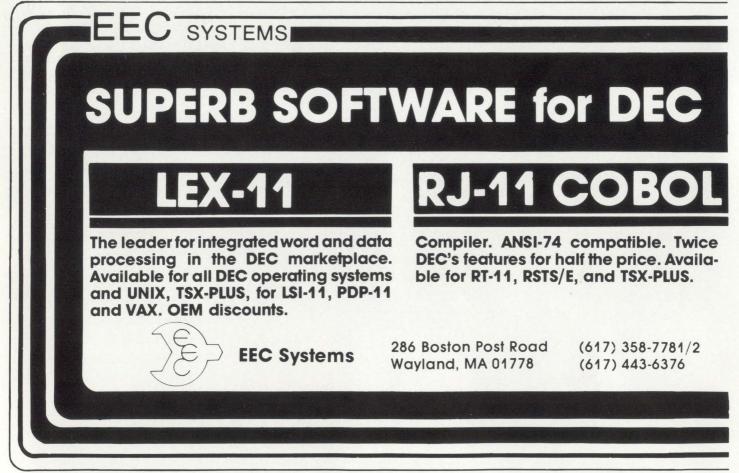
DON'T BUBBLE - QUICK SORT

By Neil Robertson, Management Controls Systems, Manchester, U.K.

It is a fairly common requirement to sort an array of numbers or strings. If the array is small this is often done using a "bubble" sort, which is possibly the slowest method available. The automatic stacking of parameters with Basic Plus functions enables a compact "quick" sort to be written. Such a function for sorting strings is given below, and also a table showing CPU seconds used when sorting random eight byte strings using Basic Plus on an 11/70. The string length is not important when using Basic Plus since the comparison is normally resolved in the first 3 or 4 bytes, and only the pointers are changed. With Basic + 2 the times are considerably longer, since the strings themselves are moved rather than the pointers.

Integer and Floating Point sorts require the string variables to be changed to integer or floating point variables, and the function name changed. The array must of course be dimensioned. Only the array elements in the range given by the first two parameters are sorted. The third parameter is used to save the variable X% for use by the second recursive function call. This function should not be used to sort virtual arrays. If you must do such a thing use a bubble sort. Purists may not like the "WHILE" syntax but it works.

! Ar	ray size ! 100 ! 200 ! 300 ! 400 ! 800 ! 1600
! Bu	bble sort-CPU sec ! 2.3 ! 9.9 ! 21.2 ! 38.3 ! 144 ! 621
! Qu	lick sort -CPU sec ! 0.5 ! 1.0 ! 1.6 ! 2.2 ! 4.7 ! 10.1
;	
4840	DEF* FNQSORT\$(Y1%,Y2%,X%)
	! QUICK SORT STRINGS X\$(Y1%) TO X\$(Y2%)
	! XX IS A DUMMY PARAMETER - SET=0
	XX=Y1X
	Y%=Y2%
	Z\$=X\$((Y1%+Y2%)/2%)
	WHILE XX<=YX
	XX=XX+1X WHILE X\$(XX) <z\$< td=""></z\$<>
	Y%=Y%-1% WHILE X\$(Y%)>Z\$
· · · · ·	IF XX<=YX THEN X1\$=X\$(XX) \ X\$(XX)=X\$(YX) \ X\$(YX)=X1\$
	X%=X%+1% \ Y%=Y%-1%
4850	NEXT
	X1\$=FNQSORT\$(Y1%,Y%,0%) IF Y1% <y%< td=""></y%<>
	X1\$=FNQSORT\$(X%,Y2%,O%) 1F X% <y2%< td=""></y2%<>



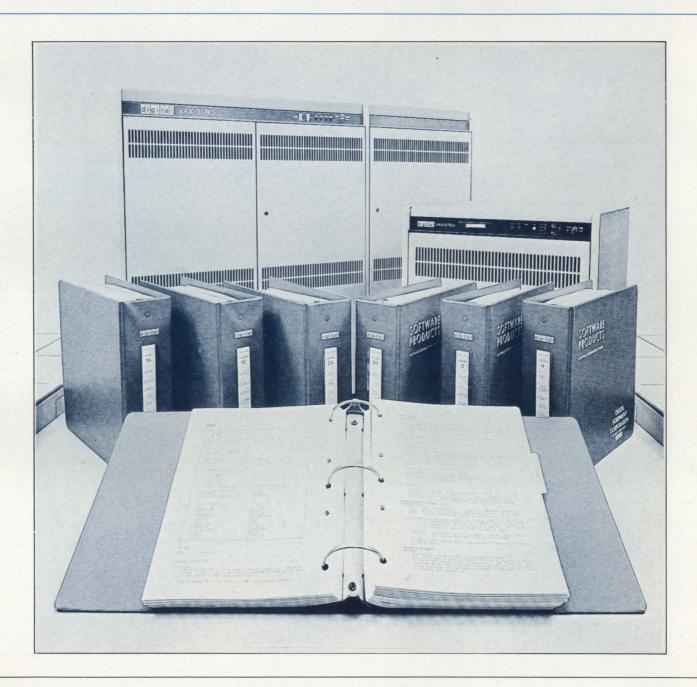
CIRCLE 43 ON READER CARD

The VAX-SCENE

Number 5

(RSTS PROFESSIONAL, Vol. 3, No. 4)

December 1981



INSIDE:

Transportable Software on RSTS/E and VAX/VMS Using a Support Library Technique

TRANSPORTABLE SOFTWARE ON RSTS/E AND VAX/VMS USING A SUPPORT LIBRARY TECHNIQUE

By Dave Froble

Transcomm Data Systems Incorporated is an Authorized DIGITAL Computer Distributor specializing in business software packages. Our experience with both DEC and RSTS dates back to 1972, when we took delivery of one of the first PDP-11/40 RSTS systems. Since that beginning, the company has developed a full line of distribution management and financial control software packages in use by clients in North America and abroad.

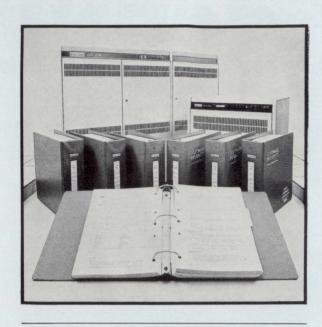
Known as TOLAS (the Transcomm On-Line Accounting System), this software was initially written in BASIC-PLUS. When BASIC-PLUS-2 (BP2) became generally available several years ago, certain critical parts of the system were converted to take advantage of the increased program size and run-time efficiencies provided by the new language.

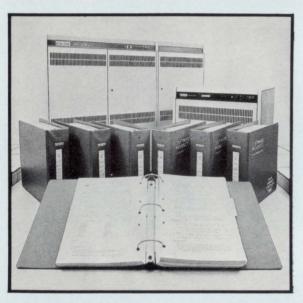
When it became clear in 1980 that Digital would be placing increasing technical and marketing emphasis on the new 32-bit VAX line of computers, Transcomm initiated a study to evaluate how best to transport TOLAS to the new operating environment. Two alternative strategies were considered. First, we could have used a compatibility mode technique. Although the actual conversion effort would have been fairly simple, the resulting software would not fully utilize the additional capabilities of the VAX and VMS. A better solution, we felt, would be to implement our software on the VAX using VAX-11 BASIC. With this alternative, the resulting software would take full advantage of the new VAX technology.

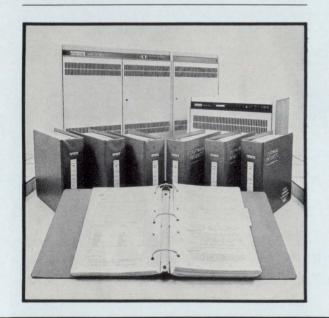
The problems associated with maintaining two sets of software (one for the PDP-11 and one for the VAX) were also considered. It was decided to avoid these problems by producing a single set of software that would run on both machines. The solution involved creating separate support libraries of external subprograms for each machine. These libraries would provide the basis for transporting common application programs between the two types of systems.

The first step in this support library technique for the VAX conversion was to translate all the RSTS/E programs into BASIC-PLUS 2. Following this step, unique support libraries were created for each type of system. These application program support routines perform the functions that are unique either to the specific computer or to its operating system. In theory, the approach could be extended to other computer systems in the future.

The first major problem we encountered concerned disk files. Disk files on the VAX are handled in a manner that differs greatly from the RSTS/E native files. Mainly, all files on the VAX are accessed using the RMS file processing







software, and RMS, as implemented on the VAX, provides excellent file handling capabilities, as contrasted with its implementation under RSTS/E. But since Transcomm had developed its data management techniques before RMS was planned, our software relied on RSTS/E block I/O techniques. VAX-11 BASIC allows block I/O simulation by opening a file with RMS ORGANIZATION VIRTUAL. The actual file organization is SEQUENTIAL with fixed length 512 byte records.

The procedures that will be discussed in this article primarily concern the handling of disk and terminal I/O. In particular, I will discuss the opening of a Data Base File, the opening of a Virtual File, the opening of an ASCII Text File, and opening the keyboard on a non-zero channel.

OPENING DATA BASE FILES BY SIMULATING RSTS/E MODE ARGUMENTS

The opening of data files in a general application support routine has been a characteristic of Transcomm software throughout our history of writing business application packages. In BASIC-PLUS, this was accomplished using a user defined function. Later, with BP2, the function was rewritten as a subprogram and linked into a task from an object library containing support subprograms. The ability to pass a parameter value to be used as the MODE argument in the OPEN statement allowed file accessing and sharing to be specified either before (via hard coding) or at run time.

VAX-11 BASIC does not support MODE with it's OPEN statement. Instead, keywords are used to define access and sharing characteristics when opening a file. Since this was unacceptable, relative to our concept of using general support routines to perform often-used and standardized operations, a simulation of MODE was required.

VAX-11 BASIC and VAX RMS did provide a possible solution. The RMS Data Structures FAB (File Access Block) and RAB (Record Access Block) are used to define file opening and record accessing operations. VAX-11 BASIC allows a USEROPEN parameter in an OPEN statement. When used, a FAB and a RAB are initialized and a user specified MACRO-32 routine is called which can modify the FAB and/or the RAB to specially define the file open. The FAB, which plays a role similar to FIRQB, is an RMS Data Block used to specify how a file is to be opened. The fields we wanted to modify are called FAC, one byte of flags used to specify access methods, and SHR, one byte of flags used to specify sharing. The RAB is a RMS data block used to control record operations. When VAX-11 BASIC passes control to a USEROPEN routine, a parameter list is passed with addresses of the FAB, the RAB, and the channel on which to open the file. The FAB is initialized, therefore, with all data necessary to open the file. The RAB has minimal data at this time. Additional data will be stored in the RAB after completion of the USEROPEN routine.

The first part of the solution was to determine the bit mask values for different specifications of the ACCESS and ALLOW keywords. A small program opening a file and using a USEROPEN routine was created and executed using ACCESS MODIFY, WRITE, and READ, and ALLOW MODIFY, READ, and NONE. Our USEROPEN routine put the FAB and

RAB values into a global PSECT (COMMON) data area and the values were displayed by the program following completion of the OPEN statement. The following values were obtained:

	ACCESS	ALLOW
MODIFY	15	15
WRITE	3	N/A
READ	2	2
NONE	N/A	32

Note that these values are represented by external equivalences, and that the equivalence mnumonics can be declared as external constants. This technique is compatible with VMS programming practices.

With this information, we modified our file open support routine to set up two 8 bit masks based upon arbitrary codes representing different combinations of ACCESS and ALLOW. Refer to the subprogram FBOPEN to see those combinations that can be used. For compatibility, a code of 1 stood for ACCESS MODIFY, ALLOW MODIFY, the equivalent of update mode under RSTS/E. A code of O is equivalent with the code for ACCESS WRITE, ALLOW NONE. A long word common block is used to pass the two byte values to the USEROPEN routine.

	SUB FBOPEN (CH	18 , FILENAMES , CBF8 , MO.DE8 ,	, ERROR& , ES , FLD&(,)) &
		- CHANNEL TO USE FOR I/O ACT	
	CH% FILE.NAME\$	- NAME OF DATA FILE &	WITY &
	CBF%		SE WHEN COMPUTING BUFFER SIZE &
	MO.DE%	- USEROPEN ACCESS AND ALLOW S	SWITCH &
		1 - ACCESS MODIFY	Y ALLOW MODIFY & Y ALLOW READ & Y ALLOW NONE & ALLOW READ &
		2 - ACCESS MODIFY	Y ALLOW READ &
		3 - ACCESS MODIFY	Y ALLOW NONE &
		4 - ACCESS WRITE	ALLOW READ &
		5 - ACCESS WRITE 6 - ACCESS READ	ALLOW NONE & ALLOW MODIFY &
		7 - ACCESS READ	ALLOW READ &
		8 - ACCESS READ	
	FRRORS	- ERROR CODE &	
	ES	- FILENAME &	
	FLD% (,)	- ARRAY TO USE FOR FIELDING I	DATA &
03!<-	<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<	*****	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
			NIGORDONIAND (
	COPYRIGHT	1981 BY TRANSCOMM DATA SYSTEMS	INCORPORATED &
	MUTC DROCRAM	IS THE SOLE PROPERTY OF TRANSCOM	MM DATA SYSTEMS &
	TNC AND MAY	NOT BE COPIED IN WHOLE OR IN PAI	RT WITHOUT THE &
		TTEN PERMISSION OF TRANSCOMM DA	
	DAFRESSED WRI	The sector of standoord in	
			>>>>>>> &
Courses			
10	1>>Append < <fi< td=""><td>:FMCOMM.B2S>> before compiling</td><td>subprogram. &</td></fi<>	:FMCOMM.B2S>> before compiling	subprogram. &
	PRINT "FMCON	M.B2S is not appended." &	
	\ STOP &		
	1		
80	COMMON (UOPEN)	LONG BUFF.ADDR	<pre>!>>USEROPEN COMMUNICATION &</pre>
00	ON ERROR GO	ro 900	! ERROR TRAPPING & ! CLEAR ERROR FLAG &
	<pre>\ ERROR% = 0% \ E\$ = FILENAM</pre>	AEC.	SET RETURN DATA &
	CBF8 = 18	155	SET DEFAULT CBF &
	X28 = (CH8<	191	SET NO ISAM FLAG &
	CHE = ABS (CH		1 SET CHANNEL VALUE &
	1 0110 1100101	,	6
	\ FAC% = 0%		I ACCESS FLAGS &
	\ SHR% = 0%		! ALLOW FLAGS &
	\ MO.DE% = 5%	IF MO.DE%=0%	! RSTS/E MODE ZERO &
			8
	\ FAC% = 15%	IF MO.DE% = 1%	1 ACCESS MODIFY &
		OR MO.DE% = 2%	6
		OR MO.DE% = 3%	6
	\setminus FAC% = 3%	IF MO.DE% = 4%	ACCESS WRITE &
	\ FAC% = 2%	OR MO.DE% = 5% IF MO.DE% = 6%	ACCESS READ &
	(FAC8 = 28	OR MO.DE% = 7%	ACCESS READ &
		OR MO.DE8 = 88	
		OR HOIDER - DE	
	\ SHR% = 15%	IF MO.DE% = 1%	ALLOW MODIFY &
		OR MO.DE% = 6%	6
	\ SHR% = 2%	IF MO.DE% = 2%	ALLOW READ &
		OR MO.DE% = 4%	
		OR MO.DE% = 7%	
	\ SHR% = 32%	IF MO.DE% = 3%	1 ALLOW NONE &
		OR MO.DE% = 5%	6
		OR MO.DE% = 8%	6
		IF FAC%=0% OR SHR%=0%	I TEST MODE VALUE 6
		ERROR®	6
	\ GOTO 999 IF		8
	\ GOTO 999 IF		
	\ GOTO 999 IF	SHR&*256% + FAC%	
	\ GOTO 999 IF \ BUFF.ADDR =		
	\ GOTO 999 IF \ BUFF.ADDR = \ OPEN E\$ FOR	INPUT AS FILE CH&,	5 ! OPEN DATA FILE 5
	\ GOTO 999 IF \ BUFF.ADDR = \ OPEN E\$ FOR ORGAN	INPUT AS FILE CH8, IZATION VIRTUAL,	6 ! OPEN DATA FILE 6 ! BLOCK I/O 6
	\ GOTO 999 IF \ BUFF.ADDR = \ OPEN E\$ FOR ORGAN	INPUT AS FILE CH&,	! OPEN DATA FILE &

December 1981 RSTSPROFESSIONAL

	\ RABADR(CH%) = BUFF.ADDR	SAVE RAB ADDRESS &
	\ CALL BUFADR \ BUFADR(CH%) = BUFF.ADDR	Set BUFFER ADDRESS & SAVE BUFFER ADDRESS
	<pre>\ FDB(CH%,3%) = -1. \ FDB%(CH%,9%) = -1% \ FDB%(CH%,9%) = -1% \ FDB%(CH%,10%) = 0% \ FDB%(CH%,11%) = CB%</pre>	6 ! SET FM DATA 6 5 5 5
	<pre>\ FDB%(CH%, Z0%) = FNZ%(20%+10%) FOR Z0% = 0% TO 7%</pre>	S READ FOB DATA S
	\ FDB%(CH%,2%) = MO.DE%	SET FM DATA &
	<pre>\ PDB(CH%,0%) = 32768.*FN2%(0%) + FN2%(1%) \ FDB(CH%,1%) = 32768.*FN2%(2%) + FN2%(3%)</pre>	6 ! READ FDB DATA 6 6
	<pre>\ FLD%(0%,0%) = FDB%(CH%,3%) \ FLD%(0%,1%) = 0% \ FLD%(0%,2%) = 0%</pre>	5 ! SET FIELD ZERO DATA 5 5
	\FOR 20% = 1% TO FDB%(CH%,5%) \ 2% = 16% + 20%*16%	6 ! READ FIELD DATA 6 6
	<pre>\ PLD\$(20\$,0\$) = FN2\$(2\$) PLD\$(20\$,1\$) = FN2\$(2\$+1\$) YLD\$(20\$,1\$) = FN2\$(2\$+2\$) AND 255 21\$ = FN2\$(2\$+3\$) 22\$ = Z1\$ AND 255\$</pre>	! LENGTH & ! OFFSET & ! DATA TYPE & &
210	<pre>\ 22% = 21% AND 255% 21% = SWAP\$(21%) AND 255% \ FLD%(20%,2%) = FLD%(20%,2%) OR SWAP\$(21% AND \ FLD%(20%,2%) = FLD%(20%,2%) OR SWAP\$((22% AND NEXT 20%</pre>	15%)*16%) &
	<pre>\ FDB%(CH%,1%) = 2% IF X2% \ 2% = FDB%(CH%,1%) \ GOTO 220 UNLESS 2%=1% OR 2%=5%</pre>	5 1 SET FILE TYPE 5 5
	<pre>\ NUM.KEYS% = FNZ2%(3%) KEYS%(CH%,0%) = NUM.KEYS% NUM.KEYS% = 5% IF NUM.KEYS%>5% KEYS%(CH%,1%) = CH% SPLIT% = FNZ2%(4%) SPLIT% = 5% IF SPLIT%<1% OR SPLIT%>9%</pre>	5 1 READ NUM OF KEYS FROM KDB 5 5 1 SET KEY ACTIVITY CHANNEL 5 1 READ SPLIT RATIO 5 1 DEFAULT TO 50/50 5
	<pre>\ KEYS%(CH%,2%) = SPLIT%</pre>	1 SET SPLIT RATIO & &
	<pre>\ KEYS%(CH%,20%*8%-4%+2%) = FNZ2%(20%*16%+1%+2%) FOR 2% = 0% TO 3% FOR 20% = 1% TO NUM.KE/S%</pre>	8 ! READ KEY DATA FROM KDB & ! FOR EACH KEY &
220	UNLOCK ICHI GOTO 999	! UNLOCK I/O CHANNEL & ! EXIT &
300	DEF* FN2%(OFF%) \ CALL FMRFDB(CH%,OFF%,VALUE%,ERROR%) \ GOTO 999 IF ERROR% \ FN2% = VALUE%	1 1>>READ FDB DATA & & & &
310	FNEND	5 5
320	DEF* FNZ2%(X%) = FNZ%(FDB%(CH%,6%)*256% + X%)	! & !>>READ KDB DATA & &
900	ERROR% = ERR \ RESUME 999	! !>>SET ERROR CODE & ! EXIT &
999	SUBEND	1

The USEROPEN routine requires the use of several RMS equivalence definitions and routines. The system macros \$FABDEF and \$RABDEF are used to set up numeric values representing offsets and values for the respective data blocks. For example, the offset from the beginning of the FAB to the access byte is represented by FAB\$B-FAC. (Chapter 4, in the VAX-11 RMS reference manual, contains information on the FAB while RAB information is contained in Chapter 5.) Processing of the USEROPEN includes saving the FAB and RAB addresses, setting the ACCESS and SHARE bytes, returning the RAB address in the global PSECT (for possible use later in the program), issuing the RMS \$OPEN call using the FAB, and issuing the RMS \$CONNECT call using the RAB. Upon exit of the routine, execution of the VAX-11 BASIC OPEN statement continues.

.TITLE USEOPEN - Useropen routine for BASIC open .IDENT "VI-01" COPYRIGHT 1981 BY TRANSCOMM DATA SYSTEMS INCORPORATED THIS PROGRAM IS THE SOLE PROPERTY OF TRANSCOMM DATA SYSTEMS INC. AND MAY NOT BE COPIED IN WHOLE OR IN PART WITHOUT THE EXPRESSED WRITTEN PREMISSION OF TRANSCOMM DATA SYSTEMS INC. Abstract: This routine can be used with a VAX-11 BASIC open statement to allow parameterization of the ACCESS and ALLOW clauses. A global PSECT is set up for communication with the calling program since the three arguments passed to a USEROPEN routine are controlled hv BASIC. This PSECT contains one longword that is used to pass bit masks for the access and share fields in the PAB, and to pass back the address of the RAB to the calling routine. Two code PSECTs are set up: USEOPEN is the PSECT that sets the FAB fields, opens and connects the channel, and returns the RAB address. JFADR is a PSECT to get the data buffer address from the AB field UBF.

: Author: Dave Froble, Transcomm Data Systems, Pittsburgh, Pa. Access field values: (ACCESS) MODIFY - 15 WRITE - 3 READ - 2 Share field values: (ALLOW) MODIFY - 15 READ - 2 NONE 32 Define FAB and RAB symbols ; FAB symbols ; RAB symbols Define global psect, used for passing data to/from routine ; .PSECT UOPEN, PIC, OVR, REL, GBL, SHR, NOEXE, RD, WRT, LONG ; RAB address Code part of routine .PSECT \$CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,LONG .ENTRY USEOPEN,^M<R2,R3> ; Save registers the Save registers that will be used Get FAB and RAB addresses MOVL 4(AP),R2 MOVL 8(AP),R3 ; Address of FAB ; Address of RAB Set FAB access and shareing flag bytes MOVB PDAT, FAB\$B FAC(R2) ; Set access byt MOVB PDAT+1, FAB\$B SHR(R2) ; Set share byte Return RAB address MOVL R3, PDAT ; Return RAB address Set RAB fields ; Do the \$RAB routine here ; Open the file here SOPEN FAB=04 (AP) ; Open file using FAB ; Test for error ; Connect to file BLBC R0,105 SCONNECT RAB=@8(AP) Return here 10\$: RET : Return with RO containing status Routine BUFADE - get data buffer address from RAB .PSECT \$CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,LONG .ENTRY BUFADR,^M<R3> ; Save registers that will be used : RAB address ; Get huffer address : Done MOVI PDAT, R3 RABSL UBF(R3), PDAT FND

A second entry point has been included in the USEOPEN.MAR subprogram. The entry point BUFADR is used to obtain the address of the data buffer after the OPEN statement has completed. It cannot be obtained during execution of the USEOPEN section of the code, since the buffer is not allocated until after the USEROPEN routine returns control to the OPEN statement.

OPENING BLOCK I/O FILES:

A Block I/O File under VMS and VAX-11 BASIC is an RMS ORGANIZATION VIRTUAL file. This file may be used for virtural arrays or to simulate RSTS/E block I/O. The routine developed for this purpose was a by-product of the research to develop the data file open routine previously discussed.

An argument to specify either the creation of a new file, or the opening of an existing file, is included in the argument list of this routine. Based upon this flag, the routine conditionally executes one of two OPEN statements. With this routine, consequently, simulation of the RSTS/E mode features are available on the VAX.

100!<<<<< OPEN AN RMS VIRTUAL FILE >>>>>>> & SUB VMOPEN (FILE.NAME\$, CH% , MO.DE% , STAT% , ERROR%) & FILE.NAME\$ - FILENAME OF FILE TO OPEN & CHANNEL TO USE IN OPEN & USEROPEN ACCESS AND ALLOW SWITCH CH% MO.DE% SS AND ALLOW SWITCH 6 ACCESS MODIFY ALLOW MODIFY 6 ACCESS MODIFY ALLOW NEAD 6 ACCESS WODIFY ALLOW NORE 6 ACCESS WRITE ALLOW NORE 6 ACCESS WRITE ALLOW NONE 6 ACCESS READ ALLOW READ 6 ACCESS READ ALLOW READ 6 ACCESS READ ALLOW NONE 6 8 - ACCESS PEAD ALLOW NO - CREATE OR OPEN SWITCH & 1 - CREATE A NEW FILE & 2 - OPEN AN EXISTING FILE & - RETURN ERROR FLAG & STATS ERROR

SPR	ROFESSIONALRSTSPROFESSIONALRSTSPROFESSIONALRSTSP	ROFESSIONALRSTSPROFESSIONALRSTSPROFE	SSIONALRSTSPROFESSIONA
<<	COPYRIGHT 1981 BY TRANSCOMM DATA SYSTEMS		830 PRINT "INVALID CHANNEL"; CH%; "IN VMOPEN" ! BAD CHANNEL & \ STOP & \ GOTO 999 & \$
	COPINIONI INT DI TRADCONT ONTA DIOTATO I		(0010 333
	THIS PROGRAM IS THE SOLE PROPERTY OF TRANSCOM INC. AND MAY NOT BE COPIED IN WHOLE OR IN PAI EXPRESSED WRITTEN PERMISSION OF TRANSCOMM DAT	RT WITHOUT THE &	840 PRINT "CHANNEL"; CH&; "IS ALREADY OPEN IN VMOPEN" & \ STOP \ GOTO 999
<<		·>>>>>>>> §	1
	COMMON (FLGS) LONG FLAG	I>>USEROPEN COMMUNICATION &	900 ERROR® = ERR ! ALL ERRORS & \ RESUME 999 &
	ON ERROR GOTO 900 \ ERROR% = 0% \ FAC% = 0% \ SHR% = 0%	! ERROR TRAPPING & ! CLEAR ERROR FLAG & ! CLEAR ACCESS FLAG & ! CLEAR SHARE FLAG &	999 SUBEND
	\ GOTO 810 UNLESS STAT%=1% OR STAT%=2% \ GOTO 820 IF FILE.NAMES = "" \ GOTO 830 IF CH%<1% OR CH%>99%	6 ! TEST 'STAT%' 6 ! TEST FOR A FILENAME 6 ! TEST CHANNEL VALUE 6	.TITLE FOPEN - Useropen routine for BASIC open .IDENT "V1-01"
	\ GOTO 840 IF BUFSIZ(CH%) > 0%	TEST FOR CHANNEL IN USE &	;++ ; **********************************
	\ FAC% = 15% IF MO.DE% = 1% OR MO.DE% = 2%	ACCESS MODIFY &	COPYRIGHT 1981 BY TRANSCOMM DATA SYSTEMS INCORPORATED
	OR MO.DE% = 3%	6	THIS PROGRAM IS THE SOLE PROPERTY OF TRANSCOMM DATA SYSTEMS
	\ FAC% = 3% IF MO.DE% = 4%	1 ACCESS WRITE &	INC. AND MAY NOT BE COPIED IN WHOLE OR IN PART WITHOUT THE
	OR MO.DE% = 5% \ FAC% = 2% IF MO.DE% = 6%	ACCESS READ &	EXPRESSED WRITTEN PERMISSION OF TRANSCOMM DATA SYSTEMS INC.
	OR MO.DE% = 7%	6	1
	OR MO.DE% = 8%	6	***************************************
	\ SHR% = 15% IF MO.DE% = 1%	& ! ALLOW MODIFY &	; Abstract:
	$\$ SHR8 = 158 IF MO.DE8 = 16 OR MO.DE8 = 68	å	; This routine can be used with a VAX-11 BASIC open statement
	\ SHR% = 2% IF MO.DE% = 2%	! ALLOW READ &	; to allow modification of the FAB and the RAB fields. In
	OR MO.DE% = 4% OR MO.DE% = 7%	6	; particular there will be parameterization of the ACCESS and ALLOW clauses.
	OR MO.DE% = 7% \SHR% = 32% IF MO.DE% = 3%	ALLOW NONE &	; and Allow Clauses.
	OR MO.DE% = 5%	ABBON NONE &	A global PSECT is set up for communication with the calling
	OR MO.DE% = 8%	6	; program since the three arguments passed to a USEROPEN
	<pre>\ ERROR% = -2% IF FAC%=0% OR SHR%=0%</pre>	& ! INSURE VALID MODE &	; routine are controlled by BASIC. This PSECT contains four
	\ GOTO 999 IF ERROR®	S INSURE VALID MODE &	; bytes. Two are used to pass bit masks for the access
		5	; and share fields in the FAB. Bit 0 of the third byte is
	\ FLAG = SHR%*256% + FAC%	! SET FLAG WORD & &	; a flag to set access APPEND.
	\ OPEN FILE.NAME\$ FOR OUTPUT AS FILE CH*,	& ! CREATE A FILE &	; Author:
	ORGANIZATION VIRTUAL, USEROPEN VFCREATE	5	Dave Froble, Transcomm Data Systems, Pittsburgh, Pa.
	IF STAT% = 1%	6	; Access field values: Byte 0 - (ACCESS)
	\ OPEN FILE.NAME\$ FOR INPUT AS FILE CH&,	! OPEN EXISTING FILE &	1
	ORGANIZATION VIRTUAL,	5	; MODIFY - 15
	USEROPEN FOPEN IF STAT% = 2%	a E	; WRITE - 3
	\ GOTO 999	6	; READ - 2
	PRINT "INVALID 'STAT%' VALUE"; STAT%; "IN V		; Share field values: Byte 1 - (ALLOW)
	\ STOP	I TEST STAT VALUE &	1
	\ GOTO 999	6	; MODIFY - 15
		1	; READ - 2
	PRINT "INVALID FILENAME IN VMOPEN"	NO FILENAME &	NONE - 32
	\ GOTO 999	6	; Append mode: Byte 2 - Bit 0

RSTS/E ON VAX ROSS/V (RSTS/E Operating System Simulator for VAX)

ROSS/V is a software package, written in VAX-11 MACRO, which provides a RSTS/E monitor environment for programs running in PDP-11 compatibility mode on DEC's VAX-11.

ROSS/V supports:

- The BASIC-PLUS interactive environment.
- Concurrent use of multiple run-time systems.
- Update mode (multi-user read/write access to shared files.)
- CCL (Concise Command Language) commands.
- An extensive subset of RSTS/E monitor calls.

ROSS/V runs under VMS and interfaces to programs and run-time systems at the RSTS/E monitor call level. ROSS/V makes it possible for DEC PDP-11 RSTS/E users to move many of their applications directly to the VAX with little or no modification and to continue program development on the VAX in the uniquely hospitable RSTS/E environment. Most BASIC-PLUS programs will run under an unmodified BASIC-PLUS run-time system.

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

ROSS/V is available from:

(Eastern U.S.) Evans Griffiths & Hart, Inc. 55 Waltham Street Lexington, Massachusetts 02173 (617) 861-0670 (Central U.S.) Interactive Information Systems, Inc. 10 Knollcrest Drive Cincinnati, Ohio 45237 (513) 761-0132 CIRCLE 67 ON READER CARD (Western U.S.) Online Data Processing, Inc. N. 637 Hamilton Spokane, Washington 99202 (509) 484-3400

;	
;	
; Define FAB and RAB symbols ; \$FABDEF	; FAB symbols
\$RABDEF	; RAB symbols
; Define global psect, used for passin ;	
.PSECT FLGS, PIC, OVR, REL, GBL, S FLAG: .LONG	; Option flags
; Code part of routine	
.PSECT \$CODE,PIC,CON,REL,LCL, .ENTRY FOPEN,^M <r2,r3></r2,r3>	SHR,EXE,RD,NOWRT,LONG ; Save registers that will be used
; ; Get FAB and RAB addresses	
; MOVL 4(AP),R2 MOVL 8(AP),R3	; Address of FAB ; Address of RAB
; ; Set FAB access and shareing flag byt	
; MOVB FLAG, FAB\$B_FAC(R2)	
MOVB FLAG+1, FAB\$B_SHR(R2) ; ; Set EOF flag in RAB field ROP	; Set share byte
; Set EOF flag in RAB field ROP ; BITB #1,FLAG+2	; Test flag byte 2, bit 0
BEQLU 10\$ BISL2 #RAB\$M_EOF,RAB\$L_ROP(R	; Skip if append flag is not set
; Open the file here	
; 10\$: \$OPEN FAB=@4(AP) BLBC R0,20\$; Open file using FAB ; "est for error
\$CONNECT RAB=@8(AP)	; Connect to file
; Return here	
20\$: RET	; Return with R0 containing status
. END	uting for BASIC and
.TITLE VFCREATE - Useropen ro .IDENT "V1-01" ;++	

COPYRIGHT 1981 BY TRANSCOMM	
; THIS PROGRAM IS THE SOLE PROPE : INC. AND MAY NOT BE COPIED IN	WHOLE OR IN PART WITHOUT THE
EXPRESSED WRITTEN PERMISSION O	
; Abstract:	
; to allow parameterization of t	he ACCESS and ALLOW clauses.
; A global PSECT is set up for comprogram since the three argume	nts passed to a USEROPEN
; routine are controlled by BASI ; longword that is used to pass	C. This PSECT contains one bit masks for the access
; and share fields in the FAB. ; ; Author:	
; Author: ; ; Dave Froble, Transcomm Data Sy	stems, Pittsburgh, Pa.
; ; Access field values: (ACCESS)	
; MODIFY - 15 WRITE - 3	
; READ - 2	
; Share field values: (ALLOW)	
; MODIFY - 15 ; READ - 2	
; NONE - 32 ;	
; ; ; Define FAB symbols	
; \$FABDEF	; FAB symbols
; ; Define global psect, used for passing	
; .PSECT FLGS, PIC, OVR, REL, GBL, S	
FLAG: .LONG ; ; Code part of routine	; Option mays
	SHR, EXE, RD, NOWRT, LONG
•	; Save registers that will be used
; Get FAB address	Address of FIR
MOVL 4(AP),R2 ; ; Set FAB access and shareing flag byt	; Address of FAB es
; MOVB FLAG, FAB\$B FAC(R2)	; Set access byte
MOVB FLAG+1, FAB\$B_SHR(R2)	; Set share byte
; Create the file here	
<pre>\$CREATE FAB=@4(AP) BLBC R0,10\$ \$CONNECT RAB=@8(AP)</pre>	; Create file using FAB ; Test for error ; Connect to file
; Return here	, connect to the
ilos: RET	; Return with RO containing status
; .END	

OPENING ASCII TEXT FILES:

An ASCII Text File under VMS is an RMS SEQUENTIAL FILE with variable length records and carriage return record attributes. There are two important considerations for programmers with a RSTS/E background when using these files.

First, a maximum record size must be specified to prevent line overflow. A line longer than the maximum will be split into two or more records. The carriage returns are not included in sequential records but, rather, inserted when the record is printed. This effect is similar to setting a terminal's width to some small number and experiencing wrap-around.

The second consideration is line termination characters. A line feed will not act as a terminator but will be included in the record as a character. A string of line feeds spacing down to a print line can cause the record to overflow. The technique we have successfully used is to print one space for each line to be skipped.

The following routine is used to open an output text file. The same arguments can be used in a RSTS/E version to enhance transportability.

100!<<<	<<<<< OPEN ASCII TEXT FILES	\$ >>>>>> &
8	SUB FAOPEN (CH% , FILE.NAME\$, FLAG% , R.SZ% , EF	
6		
i	FILE.NAMES - NAME OF FILE TO OPEN OR CREATE	
1	FLAG% - OPTION FLAGS & 1 - OPEN FOR INPUT &	
1	2 - OPEN FOR OUTPUT & 4 - OPEN APPEND MODE &	
i	4 - OPEN APPEND MODE & 8 - OPEN READ ONLY &	
1	R.SZ% - MAXIMUM SIZE OF SEQUENTIAL RECO ERROR% - ERROR FLAG &	RDS &
\$		
! 103!<<<	······	\$ ******
& 1	COPYRIGHT 1981 BY TRANSCOMM DATA SYSTEMS INCOF	PORATED &
6	THIS PROGRAM IS THE SOLE PROPERTY OF TRANSCOMM DA	TA CYCTTONC C
1	INC. AND MAY NOT BE COPIED IN WHOLE OR IN PART WI	THOUT THE &
1	EXPRESSED WRITTEN PERMISSION OF TRANSCOMM DATA SY	STEMS INC. &
	······································	>>>>>> &
180	COMMON (FLGS) WORD CW1, CW2	USEROPEN COMMUNICATION &
200	ON ERROR GOTO 900	
	\ ERROR% = 0%	CLEAR ERROR FLAG &
	\ FAC% = 3%	
	\ FAC% = 2% IF FLAG% AND 8%	READ ONLY &
	\ SHR% = 32% ! \ SHR% = 2% IF FAC%=2% !	
	\ APP% = 0% \ APP% = 1% UNLESS FAC%=2%	NO APPEND MODE & APPEND MODE UNLESS READING :
	\ CW1 = SHR% * 256% + FAC% \ CW2 = APP%	GLOBAL BYTE 3 &
	\ REC.SIZ% = R.SZ%	
	\ REC.SIZ% = 132% IF R.SZ% < 1%	SET DEFAULT = 132 &
	\ GOTO 230 IF FLAG% AND 2%	OPEN FOR OUTPUT &
220	OPEN FILE.NAMES FOR INPUT AS FILE CH4, ORGANIZATION SEQUENTIAL VARIABLE, USEROPEN FOPEN	OPEN FOR INPUT &
	\ GOTO 999 &	
	8	
230	OPEN FILE.NAME\$ FOR OUTPUT AS FILE CH%, ORGANIZATION SEQUENTIAL VARIABLE, ACCESS WRITE,	OPEN FOR OUTPUT &
	ALLOW NONE, RECORDSIZE REC.SIZ%	
	\ GOTO 999 6	
	4 GO10 999	
900	!>>ERRORS	
910	RESUME 230 IF (ERL=220% AND ERR=5%) UNLESS (FLA	CA 100 101 4
510		FILE NOT FOUND & OPEN FOR OUTPUT & UNLESS 'INPUT' SPECIFIED &
990	ERROR% = ERR	
	\ RESUME 999	
	1	
999	SUBEND	

OPENING THE KEYBOARD:

The following subprogram is self documenting. The mode argument is used only in a RSTS/E version of the routine.

December 1981

page 65 RSTSPROFESSIONALRSTSPROFESSIONA

OPEN TERMINAL ON CHANNEL &

00!<<	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	>>>>>	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
	SUB KBOPEN (KB% , KB.MODE%) &		
	KB& - CHANNEL ON WHICH TO OPEN T	HE KEY	BOARD
	KB.MODE& - OPEN MODE , NOT USED ON TH		
	DISCUSSION: 6		
	IF THE PROCESS IS RUNNING FROM A THE DESIRED TO OPEN THE TERMINAL NO A NO DUDID THE '?' DISPLAYED NOR NO NO ZEDO IN THE BASIC LANGUAGE. HOREVER, GETS IT'S INDUF FROM 'TT', BUT PRIVE THIS ONLY WORKS IF USINC CHANNEL ISE OVERLY CONCERNED WITH THE FORMAT OF BATCH JOB RUNNING A REPORT, WE CAN S SET THE I/O CHANNEL TO ZERO IF NOT R TERMINAL. THE TERMINAL NAME WILL NOT IF NOT RUNNING FROM A TERMINAL. \$	A DET. A DET. S TO RO. SIN THE LO SKIP TH RUNNING	CHANNEL TO 6 UT ON CHANNEL 6 ACHED JOB 6 SYSSOUTPUT". 6 CE WE ARE NOT 6 GFILE OF A 6 E OPEN AND 6 FROM A 6
031<<	***************************************	>>>>>>	>>>>>> 6
	COPYRIGHT 1981 BY TRANSCOMM DATA SYSTEMS	INCORP	ORATED &
	THIS PROGRAM IS THE SOLE PROPERTY OF TRANSCO	MM DAT	A SYSTEMS &
	INC. AND MAY NOT BE COPIED IN WHOLE OR IN PA		
	EXPRESSED WRITTEN PERMISSION OF TRANSCOMM DA	TA SYS	TEMS INC. &
<<<<<		>>>>>	>>>>>> 6
00	CALL SSTERMNAM (KB\$)	1	GET TERMINAL NAME &
	\ GOTO 310 IF LEFT(KB\$,2%) = "TT"	1	TEST FOR TERMINAL &
	\ GOTO 310 IF LEFT(KB\$,2%) = "OP"	1	TEST FOR CONSOLE &
	\ KB% = 0%		USE CHANNEL ZERO &

BUILDING A BORN-AGAIN 40 Another Case Study—Continued

By Dave Mallery

In our last installment, we had an 11/40 CPU with memory management and EIS in place, happily running a toggled in memory diagnostic in 128KW of CDC MOS parity memory plugged into a big 9 slot ABLE backplane. The novelty of a successful memory diagnostic wears off after a few passes, and one is left yearning for something more in life.

In this case, the next event was a tape drive from innovative technology. It's a dual density 45 IPS model, sold by CDC. The controller is a hex high TU10 emulatator and a quad high 1600 peg board made by Emulex. You switch density on the drive, not in the program. To install the hex board, all you have to do is remove the NPG jumper on the backplane. This is a jumper between CA1 and CB1 whose absence allows the device in this slot to do DMA transfers to and from memory. With a DEC backplane you need a searchlight and a magnifying glass to find the jumper, with ABLE, the jumpers are

LETTERS to the RSTS Pro continued from page 6 PHOTO CONTEST

DO YOU REMEMBER THIS?

(Photo contest, RSTS Professional, Vol. 3, #2, p.93.) Photo contests appear in the RSTS Professional occasionally and readers have until publication of the next issue to submit their answers. We may, from time to time, limit the number of correct answers eligible to receive prizes



brought out to two terminals and are the only wire-wraps on the backplane.

With the tape in place, and the addition of an ABLE Emuloader bootstrap, our diagnostic ability took a giant step. Now we could discover all we didn't know about diagnostic running. We got an emuloader because the ancient bootstrap shipped with the 40 was only able to deal with RP11 controllers (RPO2 etc) and not with the more recent RM type controllers we planned. The emuloader is a neat little card that replaces the Unibus terminator and contains a ROM full of bootstrap and diagnostic, complete with a nifty ODT that speeds up the process of toggling in.

Now we were ready for a DISC. We couldn't resist the Fujitsu Winchester offered by Systems Industries -160 MB emulating two RM02's. Ours has the 9400 controller-later versions have a single board unit. Close inspection of this unit will reveal the deep trouble U.S. manufacturers are in. It was soon installed and spinning, but no cigar. We probably had the first production unit ever shipped, so the installation documents were not up to snuff. The problem was that the diagnostic could not deter-

I believe the garbage can is for delivering large reports (too heavy for one person) from office to office. This is especially useful if it is necessary to carry reports from one building to another in bad weather, which is more often than not the case here at Plymouth State.

Jon M. Graton, Programmer Analyst Plymouth State College

P.S. Just in case ... I take a medium. We'll put a medium aside for you, Jon, ... just in case vou're right next time

My name is Brian Bates and I work for Plymouth State College, in Plymouth, New Hampshire, as a computer operator. I believe that I have the answer to your little contest. I would like to first tell you about our system.

We run a digital PDP 11/70, with two RP06 disk drives and two line printers, one is the one you have in the picture for the contest, an LP05 with the top pulled off to cut down on the number of paper jams. I would say that there are a number of reasons for the garbage can. Since we have two air conditioners this causes a tremendous current of air which does not allow the paper to stack right. It seemed rather coincidental that your contest came when it did, we have recently invested in a garbage can to allow the paper to fold correctly, since it is a pain to constantly have to go into the machine room to fix the paper.

COTO 999 310 OPEN "TT." AS FILE KRA

999 SUBEND

Using techniques like those described above, Transcomm's existing RSTS/E software has now been made available to operate on the VAX. The program routines presented in this article are solutions which Transcomm has implemented to perform operations in VAX/VMS that cannot be done using RSTS/E techniques. Any coding examples may be used by the reader for his own purposes. No responsibility for the programs can be assumed by Transcomm. Further, the disclaimer must be made that the author does not normally write assembly language routines. Constructive criticisms and suggestions are welcomed by the author at (412) 963-6770.

> mine what type of drive was out there. Now this is handled by setting a value into a register on a drive-interface card. The value was wrong we found out what it should be in memory from the diagnostic document but not the switch settings to achieve it. What followed was an hour of:

WHILE MEMORY.VALUE% <> VALUE.IN.BOOK%

- a) change a switch
- b) reset the controller
- c) examine memory
- d) groan
- e) next

At last, a value not exactly equal, but close enough to satisfy the diagnostic. Up she comes.

The final addition was on ABLE DH/ DM. This single board wonder is a far cry from the 9-slot DH11-AD it replaces. In fact, it's faster in that it does word DMA's rather than byte transfers.

Some more morals:

- 1) It's possible to assemble a RSTS engine out of 'foreign' parts that will perform amazingly well for a small investment.
- Old 11/40's never die, they just get 2) new peripherals.

I also think that if you have large reports which are spooling out it makes it much simpler to handle them. Maybe it's just a way of telling the users it's all a bunch

of garbage anyway! Sincerely, Brian Edward Bates

Plymouth State College

We're glad the crowd at Plymouth State has found answers to their stacking and handling problems. Sorry no one has found an answer to our contest. Later. Brian.

I believe the trash can in the picture is to help the paper stack correctly. This is probably due to a static electricity problem or maybe the wire basket that comes with the LP05 is too small.

> David Corbishley Laurel, Maryland

David, interesting - but not correct.

ANSWER: No one got it !! The installation is a parimutual betting system (race track). If all goes well (and it usually does), the information on pool totals and wagers is not needed, but if the two redundant computers fail they need the hard copy to manually calculate payoffs! We think that there must be a better way that might save a few trees from becoming paper. Can you think of some?

Using Sort-11 from Basic Plus-2 on RSTS/E Version 7.0

By Ron D. Troy

According to DEC manuals, it is possible to use Sort-11 via the use of call statements or their equivalent from COBOL, Fortran, or Macro-11. This use of Sort-11 from Basic has not previously been documented by DEC, but can be done without difficulty. Using Sort-11 in your own Basic program negates the need to run \$Sort directly or with an indirect command file. It allows one to use nearly any type of file structure, to run Sort within your program (invaluable in an on-line mode), and to customize the sort process to your own needs for the greatest efficiency possible. It also allows you to chain to a sort and then have the sort program chain back when finished. One could use a customized sort program or a sophisticated sort program for all sorting with the parameters passed via core common. This chain ability is available with other sorts (ex. FSORT3) and is now available with DEC's Sort-11. The sort can also be included as part of a multiple part task. Sort may not be as fast as some other sort packages, but it may be much more convenient and reliable.

Enclosed is an operational sample program for using the sort via call statements. The sequence of the calls, the arguments passed to and from the sort modules, setting up of a single 6 byte key and manipulating it are all clearly shown and explained. The sample program is set up to sort a 30 byte RMS sequential record containing a 6 byte alphanumeric key. For test purposes a 10000 record file was set up, using any method available to speed up the sort. The sort work-area size was set as large as possible, and the number of sort work-files and buffers per work file were manipulated to achieve the minimum sort time. (On a PDP 11/44 with 1 RPO-6 and 256k words core, the minimum time achieved was 1 minute, 50 seconds.) For demonstration purposes, the example given is of a full record sort, but tag (key) sorts are also possible. For large records, a tag sort would probably be faster then a full record sort. For more information on achieving maximum efficiency, see the Sort-11 manual

The sample shown does not show the use of multiple keys, or descending sorts. To sort by multiple keys, append to your work area from left to right the various keys, in the order of their importance (the most important first). Then flip the entire key and set the 'address' variable name to start at the beginning of the last word in the key. Note that the length of the entire key must be even, or a Sort error will occur. To cause a descending sort on any one key, use the change statement to create an array, subtract the value of A(I) from the value 255, and change the array back to a string. Then, append that key to its proper position in the complete key. The individual key will be given a descending sort. Note that all work on the key must be done in an area separate from the area where the record is stored.

The Sort Sequence

The first action for a sort is to open a null buffer and map out the various work areas, and call arguments. All arguments. All arguments used by the various call statements must be mapped out, including the integer variables. The various variables must then be set or cleared (ex. IWKSIZ% = 24000% as the sort work-area size). The input file is then opened and mapped out. The first call - RSORT, is used to initialize the sort. The call, as in other calls to sort, must be a "CALL BY REF (arg1, arg2, etc)." Passed in this first call are a variable for error codes to be returned (used in all calls), the keysize - which must be even and positive, maximum record size (variable length records can be sorted), key address (actually the last word of the key), sort work-area name, sort work-area size (as large as possible), the number of work files (> 3 and < 8 - try varying for greater sort speed), the number of buffers open for each file also try varying this variable). An error code of 0 returned by the sort shows that it was properly initialized.

The second major action is to read the file records and pass them to Sort. The loop ends when end of file is reached. All keys are appended into one contiguous work area. The entire key is then flipped, and is now ready to be sorted. This can easily be done using the 'Change' statement, a 'For-Next' loop, and another 'Change' statement as shown in the sample program. The call statement here has as arguments the error-code integer variable, the size of the current record, (allowing for variable length records), and the name of the area containing the record. Again, an error code of 'O' signifies proper operation.

The third action is the merge. All that is passed here is the error-code variable. Sort will return control when the merge is complete.

The fourth action is the return of the sorted records in order from Sort. First, the input file is closed, the output file opened and mapped out. The return loop has a call and a put to the output file. The call uses the error-code variable, the record size — returned by Sort, and the output record. If an error-code < 0 is returned, then all records have been returned, while a 0 indicates normal operation.

The fifth, and final operation, is to close out the sort. The only argument used is the error-code variable. The output file should then be closed, and the sort is then complete.

All calls should be done exactly as shown in the sample program. Sort will return its own error codes or RMS error codes if an error occurs. These error codes are listed in the PDP-11 Sort Reference Manual. The manual also describes the sort sequences and the key manipulation — but not for Basic. The individual words are not reversed as described in the manual; rather the entire key is reversed. The manual also has information on sort efficiency, but this information may be inaccurate. An SPR has been submitted concerning the errors or deficiencies in the manual. Note that use of the above routines is not officially supported yet by DEC, although it is reasonable to assume that such support would soon be forthcoming. Also, it should be noted that 'SORTS.OBJ' and 'SIORMS.OBJ' must be present on the system and in the account that the task build is performed. The build statement is "BUILD PROGNAME, SORTS, SIORMS/SEQ" (the SEQ for the 'RMS sequential files used here). SORTS, and SIORMS contain all the required object modules. For further information, read Chapter 5 of the Sort manual.

RSTSPROFESSIONALRSTSPROFESSIONA

The use of Sort as described above is a convenient method for easy and simple inline sorting of RMS or most other types of files. Unlike some other sorts, it is not necessary to use header records, to know how many records are being sorted, or to otherwise manipulate your file to make it acceptable to a sort, nor is it necessary to run the sort directly or through an indirect command file. You can now adapt the sort to your file, and set it up to run as efficiently as possible for your file. You can also set up a common sort program that will read nearly any type of file without problems. This is something that few, if any, other sort packages are likely to offer for RSTS/E and Basic Plus-2. It also reduces the need to buy an non-Dec sort package that might not work under future versions of RSTS/E.

MODULE FOR CALLING SORT-11, THIS MODULE CALLS THE IVARIOUS MODULES REQUIRED FOR A SORT AND INCORPORATES ITHE TYPE OF LOGIC DESCRIBED IN THE PDP-11 SORT MANUAL. ITHE ADDRESSES CALLED FOR BY THE MODULES MUST BE ON WORD BOUNDARIES (1ST BYTE OF MAPPED AREA, 3RD, ETC). ITHE RECORD MUST BE MANIPULATED IF THE FIELDS USED FOR ITHE SORT KEY ARE NOT CONSECUTIVE (HIGH FIELD FIRST). ITHIS CAN BE DONE BY MOVING THE VARIOUS FIELDS TO AN INULL DEVICE OPEN ON ANOTHER CHANNEL. ANY CHANNELS IUSED MUST BE FROM CHANNELS 1 TO 4. THE NAME OF THE IFIELD CONTAINING ONLY THE LAST WORD OF THE TOTAL KEY IS PASSED IN THE CALL STATEMENTS, ALONG WITH ITHE LENGTH OF THE KEY AND THE LENGTH OF THE ENTIRE 6000 MANUAL . THE LENGTH OF THE KEY AND THE LENGTH OF THE ENTIRE IRECORD. FOR MORE INFORMATION, PLEASE SEE CHAPTER 5. ****& ON ERROR GOTO 19000 ISET ERROR TRAPPING VOPEN 'NL: AS FILE 2%, RECORDSIZE 24200%, MAP WORKS KEY . AREA\$ = 6% . WORK.AREA\$=24000%, IERROR%, KEYSIZ%, MAXREC% . IWKSIZ%, IFILES%. IRECSZ% BIGBUF%, MAP (WORKS) FIL.2\$=4%, KEY. ADR\$=2% ::: KEY . AREAS WORK . AREAS IERRORZ = 0% ICLEAR FRROR CODE FIELD & = 0% ICLEAR ERROR CODE FIELD = 6% IREYSIZE (MUST BE EVEN) IPUT YOUR KEYSIZE HERE = 30% IRECSIZE (MUST BE EVEN) IPUT YOUR RECSIZE HERE = 24000% IWORK AREA SIZE IN BYTE KEYSIZ% MAXREC% IN BYTES& VTWKST7% ISCRATCH FILES (>2) = 4% \IFILES% BUFFER COUNT FILE 1%, JOPEN INPUT \BIGBUF% = 14 \OPEN 'SORTST.DAT' FOR INPUT = 14% AS FILE ORGANIZATION SEQUENTIAL FIXED, ACCESS READ, CLUSTERSIZE 128%, MODE 2304%, MAP SRTREC \MAP (SRTREC) IMAP OUT INPUT RECORD SORT.REC\$=30% \MAF (SRTREC) KEY.1\$=6% SORT. DAT\$=24% SORT.DA15=24% CALL RSORT BY REF (IERRORX, KEYSIZ%, MAXREC%, KEY.ADR5% WORK.AREA\$, IWKSIZ%, IFILES%, BIGBUF%) 'CALL RSORT TO INITIALIZE THE SORT 'CALL RSORT TO INITIALIZE THE SORT 'GOTO 5900 IF IERROR% <> 0 'GOTO IF ERROR OCCURED \IRECSZ% = 30% 'RECSIZE SAME AS MAX &

6020 ISECOND CALL GET #1% IREAD NEXT INPUT RECORD AT LINE 19000+ INCLUDE AN EDF (ERR=11) TEST WITH THE RESUME AT 6030 TO GO ON TO THE NEXT STAGE OF THE SORT. ISTATEMENT. \RSET KEY.AREA\$=KEY.1\$ 1+KEY.2\$+KEY.3\$, ETC. ***** \CHANGE KEY.AREA\$ TO A \B(J%)=A(KEYSIZ%-J%+1%) FOR J% = KEYSIZ% TO 1% STEP -1% \B(0%)=A(0%) CHANGE & TO KEY.AREAS \CALL RELES BY REF (IERROR%, IRECS2%, SORT,REC\$) ISORT \GOTO 6020 IF IERROR%=0% INEXT RECORD IN \GOT0 6910 GOTO IF ERROR 6030 LEND OF MAIN SORT LOOP CLOSE 1% \CALL MERGE BY REF (IERROR%) \GOTO 6930 IF IERROR% <> 0 CLOSE INPUT FILE CALL THE MERGE GOTO IF ERROR OPEN 'SORTST.SRT' FOR OUTFUT AS FILE 3%, 'OPEN OUTFUT 6040 ORGANIZATION SEQUENTIAL FIXED, ACCESS WRITE, ALLOW NONE, CLUSTERSIZE 128%, FILESIZE 70 CONTIGUOUS, 700% MAP OUTREC OUT.REC\$=30% 6050 OUTPUT WRITING CALL RETRN BY REF (IERROR%, RRECSIZ%, OUT.REC\$) !RRECSIZ IS SIZE OF THE RETURNED RECORD \IF IERROR%=0% THEN PUT \$3% \GOTO 6050 6055 GOTO 6060 IF IERROR% CO IEND OF FILE IN SORT \GOTO 6940 !ERROR ICLOSE OUT ROUTINE 6060 CLOSE \$3% \CALL ENDS BY REF (IERROR%) \GOTO 6950 IF IERROR% > 0 \GOTO 32000 GOTO IF ERROR ICLOSE FILES AND END 6900 PRINT 'INITIALIZATION '; \GOTO 6990 PRINT 'READ ': 6910 GOTO 6990 PRINT 'MERGE '; 6930 \G010 6990 5940 FRINT 'OUTPUT '; \GOTO 6990 FRINT 'CLOSE '; 6950 GOTO 6990 FRINT 'ERROR IN SORT MODULE 'FIERROR' 6990 STOP RESUME 6030 IF ERR=11 VEND OF INPUT FILE \PRINT ERR; AT ';ERL \ STOP \ RESUME 32000 19000 IEND OF PROGRAM 32000 CLOSE 1% FOR 1% = 1% TO 12% ENT

SPECIAL OFFER

MACHINE READABLE 800 BPI DOS LABEL OF MAJOR PROGRAMS IN THIS ISSUE

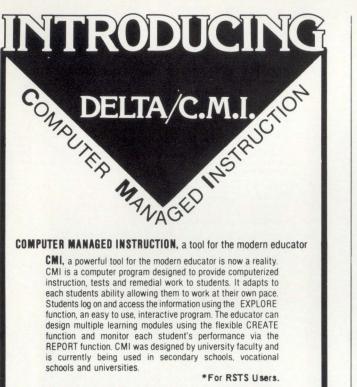
Part of the proceeds will be going to the authors.

Send \$50 to:

M SYSTEMS, INC.

Box 361, Fort Washington, PA 19034-0361

RSTSPROFESSIONALRSTSPROFESSIONA



To find out how CMI can benefit your institution contact

ICILICX The Effective Learning Exchange P.O. Box 14, Medford, New Jersey 08055 609/654-1100 CIRCLE 68 ON READER CARD

TALENTED. SHARABLE. RESPONSIVE WORD PROCESSING.

WORD-11

WORD-11 is sophisticated word processing—with features like list processing, automatic spelling error detection, indexing and footnoting—yet it's easy to learn and use.

It's sharable, up to sixty terminals. It's flexible. It's relatively inexpensive. And it's been operating for four years in hundreds of installations around the world. CIRCLE 69 ON READER CARD Available on RSTS/E, RSX and VAX/VMS versions, for DEC's family of mini-computers. Call or write: **Corporate Office:** 181 W. Orangethorpe Ave., Suite F, Placentia. CA 92670 (714) 993-4160. New York Office: (212) 687-0104. Washington, D.C. Office: (301) 657-4098.

Data Processing Design, Inc.

Optimizing Space Allocation of Literals in Basic-Plus-2

By Brad Smith, Allied Data, Olympia, WA

The author has worked on PDP-11's for 5 years in several languages. He now specializes in the design and optimization of BASIC-PLUS-2 application systems.

When a BASIC-PLUS-2 is compiled, space is allocated for every literal and variable that occurs in the program. Being aware of the differences in how literals and variables are stored can help in fine-tuning a program.

STRINGS:

Literals and variables are stored in a similar manner, with a 1-word pointer to the data, a 1-word length counter, and the data. However, literal data are stored at compile time, while only the header (pointer and length counter) are allocated for variables. Furthermore, if the same literal occurs more than once in a program, it is still stored only once. This feature can be useful if a program contains a number of similar literals which can be modified so that they are identical. For instance, if a program uses 'ABC' in one place and 'ABCD' in another, then using 'ABCD' in both places saves 8 bytes (4-byte header, 3 bytes of data, and in this case, 1 null byte because string literals always occupy an even number of bytes).

FLOATING-POINT:

Space for both literals and variables is allocated at compile time, and floating-point literal data are stored in the same areas, and interspersed with, string literals. As with strings, a floating-point literal is stored only once no matter how many times it occurs in the program, so it may be possible to save space by reducing the number of different literals. For instance, if a program contains these two conditionals—

IF A > = 100, and IF A < = 99.

—then 4 or 8 bytes (single or double precision) can be saved by using

IF A > = 100, and IF A < 100.

INTEGERS:

Integer variables are allocated at compile time, similarly to string and floating-point numbers, and an integer variable operand is actually a pointer to the datum itself, also similarly to floating-point numbers. (A string operand points to the header, which itself contains a pointer.) However, an integer literal is stored in line at compile time: that is, the operand is the number itself. Thus, an integer literal occupies less space than a variable and undoubtedly will be processed more quickly.

In many programs, these suggestions would not save an appreciable amount of space. But with that crucial program that takes a shoehorn (or a sledgehammer) to fit in memory every time you add a few lines of code, checking literal usage may well be worthwhile.





DIRECT'S ANSI COMPLIANT TERMINALS STAND APART





DI.RECT (di rekt', di-) adj. 1. honest and to the point, straightforward, frank 2. a distinctive family of ANSI compatible video display terminals, each noted for its emphasis on human engineering, functional elegance, and VT100* and VT52* compatibility 3. by the shortest way, straight 4. a family of terminals that spans the needs from interactive operation to editing and block mode and culminates with full user programmability and data entry capabilities.

For more information contact US...DIRECT!

DIRECT INCORPORATED

1279 Lawrence Station Road Sunnyvale, California 94086 (408) 734-5504 (800) 538-8404 Outside CA Telex 172277

DIRECT (U.K.) LTD.

12 Rufford Court Hardwick Grange, Woolfston, Warrington WA1 4RF Padgate (0925) 814072 Telex 666910

*Registered Trademark of Digital Equipment Corporation

VTEDIT.TEC

By David Spencer, Infinity Software Corporation

The following is a continuation of David's article which appeared in the previous issue of the RSTS Professional (Vol. 3, #3).

_ 1

"I have provided the new VTEDIT macro in a "run" able source program form. I realize that everybody is going to want to stick in some extra feature. So, it is left as an exercise to the student to souish VTEDIT down to a smaller size. (The source program runs in 10K, and can be cut down to at least 7K, and perhaps 6.) Infinity Software Corp. makes no guarantee for the performance of VTEDIT.TEC, nor takes any responsibility for the use of VTEDIT.TEC. However, inquiries, suggestions, bugs, or whatever can be sent to: David Spencer, Infinity Software Corp., 2210 Wilshire Blvd., Suite 801, Santa Monica, Calif. 90403, phone: (213) 820-2702.'

W% Restore direction flag WS Ungelete Word register Copyright (C) 1981 by Infinity Software Corporation The register P\$ is used as the substitute buffer in the GOLD ENTER sequence. It is up to the user to place text in register P\$ This soltware is solely for non-commercial use and may be only used and/or copied with the inclusion of this notice. No title to or ownership of the soltware is hereby transferred. Load get numeric argument macro The intermation in this software is subject to change without notice and should not be construed as a commitment by Infinity Software Corporation. 0°U0\$ 0U3 .U7 ZJ .U8 Q00I// < ^TU0 Q0"D Q00I// > ' Q8J \U9 ^SD Q7J ******* toad the word delimiter list ! !!VTEDIT.TEC V01.02!! Author: David Spencer - 19-Jun-81 32@^U1// 9:@^U1// 13:@^U1// 10:@^U1// 44:@^U1// WARNING: This VTEDIT destroys the numeric registers. Load control character mask Registers used: @^U2/.A.CDEF.H.JK.M....R..UVWXYZ[\.../ Character read from KB Accept numeric argument Current direction Word delimiter list Maskeu character Control character mask GoLD key flag Keypad character mask State of exit COLD character mask COLD character mask Command entered flag Enter TECO command/search string Current cnaracter position TECO command entered Scratch Scratch Scratch Scratch Scratch Diglay HELP frame Undelete direction flag Undelete direct Character read from KE :@^U2/ :@^U2/ :@^U2/ :@^U2/ **** Load keypad character mask @^U3/.../ @:TU3/.../ @:TU3/.ABCD....M..PQRS.../ @:TU3/..../mn.pqrstuvwxy.../ 48 45 55657788999 ***** Load GOLD character mask €^U4/..../ @:°U4/.../ @:°U4/.../ @:°U4/.../ @:°U4/.../ ****** Old ET value This VTEDIT macro Restore direction flag Underete Line register 1% 1\$ Load the input text macro + ! L% ! LS # U⊃3 122 OU5 .U7 ZJ .U8 155°T @°A/[H/ 155°T @°A/[K/ 10°T 155°T @°A/[K/ 155°T @°A/[H/ @:^U8/: / :G8 [8 @1/G/ O2\ _-2,.X8 -2D M8]8 Q8-."N Q8,.T TUU Q0-127°N Q0-27°N 155°T @°A/[H/ 155°T @°A/[K/ :G8 Q8,.D ' ' @D1A! ' IAI QO-127*= .-Q8*N R 0A- 9*= 155 T @ A/[H/ :G8 08,.T D F< ' 0A-10*= 155 T @ A/[A/ D F< ' 0A-13*= 155 T @ A/[A/ D F< ' 8 T 32 T @ T 0A-23* 0A-23* N & T 32 T & T ' ' D ' F< '</pre> Q7J 32768W -3W Q5 toad the HELP macro Endnyt Del L Gold Help Find Ung L Del W Page Sect Append Command |Nxt Page Una W Replace Advance | Backup Cut Dei C Bottom Тор Paste Una C Word Eol Char Enter Chngcase | Del Eol Specins Line Select Subs Open Line Reset

Hit any key to continue # @^UI\$ [C [L [W 0U6 [6 0U6 1U1 0UC 32@^UC// ETUI 2#4#8#256#512#32768ET 2****235**12*32/00E1 4,0:W 155^T @^A/</ 155^T @^A/=/ 2*= ETU7 ET#32ET ^T*< Q7ET M9 ' Q7ET ' 32768:ws127U0 1U9 0U3 Q0-127*= Q9< -.; R 0A@^UC// D > 0UC F< ' Q3*= Q0Q2U2 @01^EU2C1 | Q0Q4U2 @01^EU2GC1 ' ^TU0 Q9<1381// 10E1// > F< ^TU0 Q0^D M0 @01A1 ' ^TU0 Q0Q3U2 Q3*= @01^EU2K1 | @01^EU2GK1 ' !A! IMCI 1PK1 -1U3 ^TUO Q0-27"= @01[C1 ' Q0"D M0 ' @01A1 1PK1 Q1"> Q9L F< ' HC1 0^Q"N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q"N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q"N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^[0L ' Q9< -L > F< IAC1 0^Q N -1%9 ^ 01*,09L P< ' 0°TN -149 (10.' 09< -L > P< 01< 001RC1 '; 05/N*EGL/; R > ...= ZJ ' 0U6 F< 09< 09< -.; R 0-8/N*EGL/; R 0-8/*EGL/; > 0U6 F< IqKI IFCI !rGK! [DC! -.2"N Qy< .U7 L .-2"N 2R ' .U8 Q7J .,Q8XL K > .-2"N 1301// 1001// 2R ' ' -1UL P< Qy< .U7 -.; R :0-5/^N^EGI/*S R ' :0-5/^EGI/*U J ' Q7,.XW Q7,.D > OUW PC OUT P< Q9< 0001// > P< -0909 U7 .U8 OL Q8-.< 0A-9*= (Q7/8+1)*8U7 | %7 ^[' C > 091. IJC! 1UC1 AK IBK! Q9L Q6U9]6 Q9-Q7"N Q7U6 ' 0U8 < _-Z; 0A-13*N 0A-9*= (08/8+1)*8U8 | %8 ^[' C 08-06; > ' 06-08*N 0A-13*N R -1%8 ^[' ' [6 08U6 F< 01<< @01DE1 '</pre> Q1"< @01DK1 ' .+Q9-Z"< Q9C | ZL ' 0U6 F< .-Q9"> Q9R | J ' 0U6 F< F< ICK IDK IMK IRK ISK P< .U7 Q9*Q1:85/°EQ7/"U Q7J @OIERRI ' F< O9< XL K → 1UL F< Oy< .-z; OA€'UC// D > 1UC F< O9< .U7 :85/°EG1/"U ZJ O; ' :85/°N°EG1/"S R | ZJ ' Q7,.XW Q7,.D > -1UW F< 11K !mK! imki Q9<.U7 :PS/²EG1/¹U ZJ 0; ':PS/^N EG1/¹S K | ZJ ² ¹V; AR ¹V; A 4:WU7 Q7*N Q9< QA*A QA ISGKI Q901// F< ItGKI ZJ F< IAGKI JF< IAGKI JF< IAGKI -1*09U9 IBGKI 24*Q9L F< IXCI 0U2 @0IREGI IXCI UVEI -100 -1U2 @0!REG! IVK1 IYKI IXGCI IREGI 1U2 1U2 0U0 4:WUJ 07"N -1%7 ^[^TU8 Q8"R @^U8// Q2"> 0:^U8/:0/ Q8:0^U8// 0:^U8/U0/ ' 0:^U8/:0/ 0:^2> 0:^U8/:/ 0:U8/X/ Q8:0^U8/. Q8:0^U8/. 0:0^T< 07J ' 02"N -(:0/ Q8:0^U8// 0:^U8/-Q0)D ' 0,4:W/ M8 | 00IERRI ' | 00IERRI ' F< IVGKI IECI ^TUT Q7"R .UB Q3"= @I/M/ | @I/G/ ' Q7@I// Q8,.X8 -2D Q9< M8 > | @OIERI ' F< IwGKI @'UB/TECO Command/ 6M5"N 27@:^U6// Q9< M6 > ' F< ICC! IZC! OU4 @^U8/Returning control to TECO/ 0; IGC: IQGC: -1U4 @^U8/Aborting edit/ 0; !ERR 1 CGK 1 1 DGK 1 1.K! I.GC1 I.GK1 7°T 32768W F> IQK1 IQGK1 M9 . OIET 155°T @°A/(23;1H/ 155°T @°A/(J/ 10°T :G8 @°A/.../ 155°T @°A/>/ Q4°N 155°T @°A/(221/ Q4°> EC 13°T 10°T EX | HK EK 13°T 10°T EX ' ' 13°T 10°T [6 | C] L |W @EI// MI \$\$

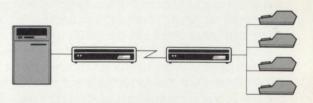
Minicomputer users:

Timeplex offers a one-stop system solution for communicating with multiple remote terminals. Economically.

Asynchronous statistical multiplexer

Synchronous statistical multiplexer

High speed modem



The Timeplex E/SERIES is a complete data concentrator system designed to economically link clusters of remote terminals to your minicomputer.

E/SERIES: Cuts communications costs.

Suddenly, saving communications costs by linking several terminals to one shared telephone line becomes easy.

Unlike the competition, the Timeplex E/SERIES simplifies the challenge of pointto-point communications by incorporating three functions in a single compact unit. One system offers you a statistical multiplexer supporting 4 to 16 asynchronous channels, *plus* an optional statistical multiplexer for an additional synchronous channel, *plus* an optional integral high speed modem. **E/SERIES: Puts it all together.** Putting three functionally distinct modules in one enclosure eliminates external communications units and bulky, expensive cables. And, a minicomputer interface option further reduces costs. The result: System planning and installation is extremely simple. Reliability is enhanced. Costs are dramatically reduced.

Free step-by-step Guide. This easy-tounderstand booklet contains all the facts on how to remote your terminals, simply *and* economically. Just write or call Timeplex for your free copy.

For the name of the E/SERIES stocking distributor nearest you, call 201-368-0736.

Timeplex, Inc./One Communications Plaza/ Rochelle Park, N.J. 07662.



The technology leader in data communications

CIRCLE 71 ON READER CARD

THE RSTS/E SYSTEM MANAGER

By Jeffrey R. Harrow, 485 Creekview Dr., Stone Mountain, GA 30083

Welcome to the anniversary issue of the RSTS Pro. I've taken a moment to look back through these issues and I'm **really** pleased at the amount of directly applicable information which I've extracted from the many articles I've read and is not available from any other source. Carl and Dave, here's one individual who appreciates what you're doing and hopes that it continues and grows.

Last issue I left you with a somewhat cryptic indication that there was a problem with using the TU77 tape drive. It appeared that the drive could write data on the tape which in reality didn't look like the data that it **thought** it had written, yet the system **did not** catch this error when the drive/ formatter combination did its "implicit read after write."

Let's take a brief look at what I mean by an "implicit read after write":

The path the tape follows begins on the supply reel, goes past the Erase Head, then the Write Head, and finally past the Read Head on its way to the take-up reel.

During a "Write" operation, the Erase Head puts the tape in a "no data" condition, the Write Head writes the data onto the tape, and then the Read Head (even though we are in a **write** operation!) reads the data which was just written by the Write Head. This data is checked for validity via parity and other checks in the TMO3 formatter.

Under normal conditions, if the data just **read** does not compare against what the formatter **thinks** that it just told the drive to **write**, the formatter flags an error condition to the MM Driver software (part of RSTS/E) which attempts to re-write the block of data using a number of industry accepted techniques (such as writing a Long Inter-record Gap over the "bad" section of tape and trying again).

Therefore, you should almost **never** be able to write a tape which you feel is "good" and is not later readable on, at least, the same drive.

This problem came to light when I began noticing that, during the COMPARE phase (which is another full Read of the tape) of BACKUP (you **do** use a COMPARE, don't you?) I was occasionally getting UNEQUAL COMPAREs on blocks of some files which I was able to verify had not changed during the BACKUP procedure. Additionally, I would usually see BAD BLOCK ON BACKUP VOLUME (ON COMPARE) errors at the same time (indicating that the drive could not read a block on the tape).

An investigation of the error log indicated that we were **NOT** taking any correctable "write" errors, much less UNcorrectable "write" errors while this tape was being written, yet, just minutes later, the same tape could not be read even after RSTS/E (the MM Driver) did its 16 retries! Further tests indicated that, indeed, the data in these blocks on the tape was not invalid.

To make a **very** long investigation short, DEC determined that there was a problem in their technical instructions for setting the Ramp-up speed of the tape drive's capstan (and are correcting this problem). Furthermore, the software group found a "hole" in the MM Driver which they felt would account for this problem and sent a software specialist out to test the patch.

The software specialist had Field Service intentionally misadjust the tape drive and reproduced the error condition. He then installed the patch **and it didn't resolve the errors**, so he and his patch went away.

About a month later, after I had experienced the problem several more times, DEC finally indicated that they **still** felt that the patch (which is going to be published in the Software Dispatch) would cure the problem and that it didn't work when the tape drive was intentionally misadjusted because of the magnitude of the misadjustment. In any event, the patch is now installed in my operating monitor, and I'm waiting to see if it reoccurs.

I have just two reservations at this time: first, what I seem to be hearing is that the error detection/recovery process should only work if the drive isn't **too** broken; and second, I have provided DEC with several instances of documented error logs where the same thing has happened on a different type of tape drive (TE16), with a different formatter (TMO2), using a different software driver (MT), on a different type of PDP-11 . . .

I'll keep you informed and still recommend that you utilize a COMPARE whenever possible!

On another subject, Bob Nixon from England wrote to me with a suggestion for providing system protection for the Basic Primer CMI package. With the executable program in the user's account given a protection code of <232>(Temporary Privilege) and the data file in the user's account (which that program reads) given a protection code of <63> (no Read, Write, or Delete access to the owner), the user should not be able to modify the data file (and hence change the name of the program to which the executable program will later CHAIN with Temporary Privilege).

Unfortunately, while the owner of a file can't do any of these things to a file with a protection code of <63>, he can RENAME the file to a protection code of <60>, and then perform whatever modifications he desires, thereby seriously breeching the system's security.

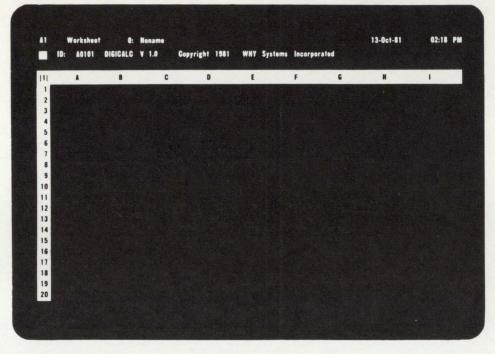
Bob, thanks for your input and I would like to hear from anyone who is interested in these topics. — See you next issue.

REPRINT POLICY

All content in this publication is copyrighted material. 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 quotations available on request.

WHY SYSTEMS

DIGICALC



THE ELECTRONIC SPREADSHEET FOR DEC COMPUTER SYSTEMS

APPLICATIONS

Financial statements Business forecasting Resource management Investment analysis Job costing Performance analysis Cash Flow analysis Error analysis

EASY TO USE

Built-in automatic training procedure Instant HELP available at the terminal User-friendly / interactive

SAVES MONEY

Reduces demand on information system personnel Eliminates long hours of "what if" calculations Report and form printing costs reduced or eliminated Low, one-time investment

PROVIDES INSTANT REPORTS Choice of formats

Working copies or board-room quality

- AUTOMATIC CALCULATION
- EXTENSIVE MATH FUNCTIONS ALGEBRAIC LOGICAL FINANCIAL SCIENTIFIC USER DEFINED FUNCTIONS
- TEN KEY NUMERIC DATA ENTRY
- EXTENSIVE HELP AT TERMINAL
- INCLUDES SELF TEACHING MODE
- WORKSHEET CONSOLIDATION
- VARIETY OF "BOARDROOM" QUALITY REPORTS
- SAVES AND RECALLS WORKSHEETS

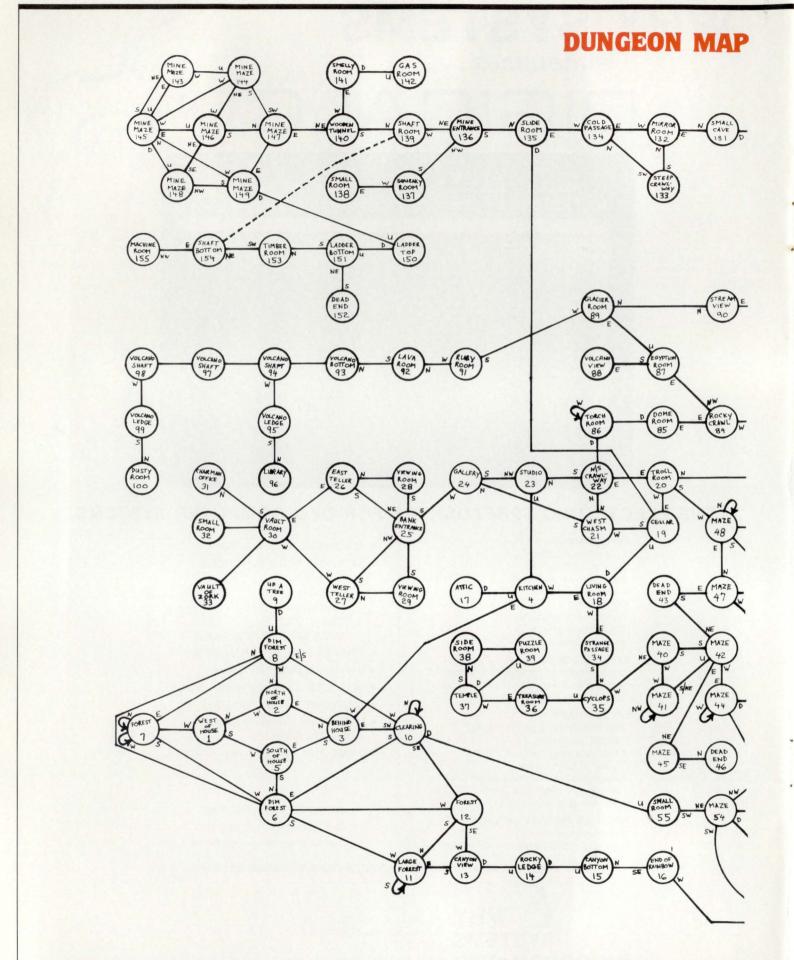
ON RSTS/E, RSX-11M, VMS SOON ON RT-11, TOPS 20 VT-100 OR MOST EMULATORS

NOW AVAILABLE AT INTRODUCTORY PRICES THRU DECEMBER 31, 1981

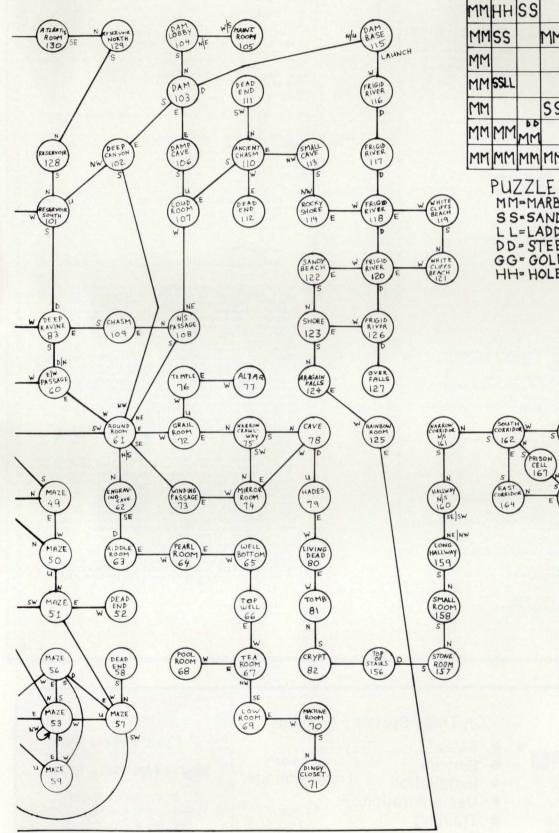
SYSTEMS 17130 Avondale Way, N.E. Redmond, Washington 98052 (206) 881-2331

INCORPORATED CALL TODAY FOR A DIAL-UP DEMONSTRATION

CIRCLE 72 ON READER CARD

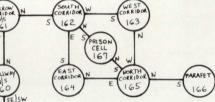


By Brian Lomasky, 528C Shennecossett Road, Groton, CT 06340



MM	MM	MM	MM	MM	MM	MM	MM
MM	HH	SS			SS		MM
MM	SS		MM		LLSS		MM
MM					MM		MM
MM	SSLL			SS GG	SS		MM
MM			SS				MM
MM	MM	MM				MM	MM
				MM	MM	MM	MM

PUZZLE ROOM MM-MARBLE WALL SS-SANDSTONE WALL L L=LADDER DD=STEEL DOOR GG=GOLD CARD HH=HOLE IN CEILING



10

CALLER.BAS

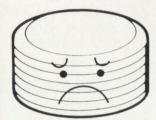
By Steven Fahey, Teachers Service Organization, Willow Grove, Pa.

ERROR PROCESSOR

CALLER was designed to help take care of small buffer problems associated with RSTS/E Version 7.0 . IF ERR = 57 AND CCL.COMMANDS = "HELP" THEN RESUME 32700 19005 Since each CCL command takes up one small buffer, having many CCL's on the system could end up causing a lot of unnecessary small buffer problems that arise when many jobs are on the system at IF ERR = 57 THEN PRINT CHRs(155%) + "H" + CHRs(155%) + "J" 19010 PRINT "Not a valid 'Call' command" PRINT "Type 'Call help' for help" the same time. V PRINT CALLER uses one CCL and eliminates the need for RESUME 32700 any others. . 1 NOTE -- STATEMENT 1 OF THIS LINE IS CLEAR SCREEN FOR VT52 Simply use the CCL command the same as you always would, but add the word 'CALL' to the front of it. 19015 IF ERR = 5 AND ERL = 2040 THEN PRINT PRINT For Example: PRINT "Can't Find "; PROGRAMS PRINT To Que the file 'TEST.LST' to the line printer, Type 'CALL QUE LPO:=TEST.LST' To Set the speed of a terminal to 2400, Type 'CALL SET SPEED 2400', And to see the valid call commands, Type 'CALL HELP'. PRINT N RESUME 32700 19099 PRINT PRINT "ERROR "; ERR; " ON LINE "; ERL V PPTNT 20 Programmer : Stephen Fahey Installation : Teachers Service Organization, Willow Grove, PA 30000 DATA STATEMENT COMMAND, DESCRIPTION, PROGRAM, LINE . TO CHAIN COMMAND, DESCRIPTION, PROGRAM, LINE + TO CHAIN
COMMAND, DESCRIPTION, PROGRAM, LINE + TO CHAIN
ATTACH*, "Attach to Detached Jobs*, "slogin,TSX*, Jogoo,
"SPC*, "BASIC PLUS Cross Reference", "SPECREF.MAC., JOGOO,
"CANCEL, "QUE/11 Cancel Verb *, "SPLICANCEL.TSX*, 0,
CHECK*, "Personal Checking", DRI:[101]CLESECSAV*, 0,
CHESS, "Chess Game*, "RRI:[100,100]CHESS.SAV*, 0,
CLES, "Clear-Screen Command*, "[1,101]CLESAC*, 0,
DIF*, "Directory Listing", "[1,6]DIR.TSX*, JOGOO,
"ED2*, "DEC Editor", "SED2*, 0,
"FINANCE", "Personal Finance", "[1,105]OPTION.MAC*, 0,
POTRAN*, "FORTAN Complier", "SFORTA*, 0,
LINK*, *LINK Program*, "SLINKF,SAV*, 8208,
LINK*, *FORTAN Complier", "SFORTA*, 0,
"LINK*, "FORTAN TOTHER PACK, 100,
"ACCO, "MACRO Program*, "SILNKF,SAV*, 8216,
"MACRO, "MACRO Program*, "SUMOUNT.BAC*, JOGOO,
"OPT.", "Option Program*, "SUMOUNT.BAC*, 0,
"OPTION*, "Option Program*, "[1,3]OPTIOP.TSX*, 0,
"OPTION*, "Option Program*, "[1,3]OPTIOP.TSX*, 0,
"OPTION*, "Option Program*, "[1,3]OPTIOP.TSX*, 0,
"SACAL, "RACKD Program*, "SLINKF, SJOOO,
"STARTEK*, "SAVER Mackup System*, "[1,1]SIGCHEP.BAC*, 30000,
"SET, "Terminal Characteristics," STISET.BAC*, 30000,
"SET, "Terminal Characteristics," STISET.BAC*, 30000,
"SCALS, "GUE/11 QUE PRINTERF*, SPL:SUMMIT.TSX*, 0,
"SAMER*, "Startrek Game*, "DRI:[100,100]TRES2_BAC*, 0,
"STARTEK*, "Startrek Game*, "DRI:[100,100]TRES2_BAC*, 0,
"STARTEK*, "Startrek Game*, "DRI:[100,100]TRES2_BAC*, 0,
"SUMMIT*, "QUE/11 QUE PRINTERF*, SPL:SUMMIT.TSX*, 30000,
"ST*, "System Status Display." "SISTAT.TSX*, 30000,
"ST*, "System Status Display." SISTAT.TSX*, 3000 1000 ENTRY PROCESSOR 30010 EXTEND ON ERROR GOTO 19000 1010 CET.CORE.COMMONS = SYS(CHRs(7%)) \ GET.CORE.COMMONS = MID(GET.CORE.COMMONS, 6%, LEN(GET.CORE.COMMONS) - 5%) 1030 IF END.OF.CCL.COMMAND: = 0: THEN CCL.COMMANDS = GET.CORE.COMMONS GOTO 2010 1040 1050 CCL.COMMANDS CVTss(MID(GET.CORE.COMMONS, 1%, END.OF.CCL.COMMAND% - 1%), 4%)& 2000 MAIN PROGRAM PUT.CORE.COMMONS = SYS(CHR8(8%) + GET.CORE.COMMONS)
\ IF CCL.COMMANDS = "HELP" THEN
PRINT CHR8(155%) + "J";
PRINT CHR8(155%) + "J";
PRINT "Command", "Description", ". TIMES(0%); DATES(0%)
PRINT "______" 2010 PRINT 2020 READ CCLS, PROGRAM.DESCRIPTIONS, PROGRAMS, LINE.NUMBER: £ 2030 IF CCL.COMMANDS = "HELP" THEN PRINT CCLS, PROGRAM.DESCRIPTIONS \ GO TO 2020 32700 IF CCLS <> CCL.COMMANDS THEN GOTO 2020 ELSE CHAIN PROGRAMS LINE LINE.NUMBER: 2040 END PROCESSOR 19000 1 32767 END



OPTIMIZE your RSTS/E disks with DSKBLD.



FILING PROBLEMS YOU MAY BE PUTTING UP WITH.

• Fragmented UFD's

• Poor file clustersizes.

- Excessive arm movement.
- Excessive FIP overhead.
- Disk bound response problems.

DSKBLD HAS THE SOLUTIONS.

- Contiguous UFD's.
- Organized files.

Scattered files.

- Minimized FIP overhead.Reduced_arm movement.
- Optimum clustersizes. Lower disk access overheads.

Overtime for disk management is greatly reduced or eliminated. Disks get rebuilt more often because it's painless.

DSKBLD is an extremely fast RSTS/E disk-to-disk utility which:

- \Box Supports mixed disk types.
- □Untangles directory links.
- □Creates contiguous UFD's to proper size.
- □ Creates accounts in sorted order.

- □ Locates files adjacent to owner's UFD.
- \Box Optimizes file clustersizes.

*RSTS/E is a registered trademark of

- \Box Saves accounting data.
- □ Is easy to use and fully documented.
- Provides logging and statistics.
- \Box Is inexpensive.

INTRODUCTORY PRICE \$500.

		Digital Equipment Corporation.
	HOME OFFICE CONTACT: Gary Vowels	Send me more information on the DSKBLD RSTS/E Disk.
	1700 Westlake Avenue N. Lake Union Building	Name
	Seattle, Washington 98109	Title
SERVICES CORPORATION	206-285-3260	Company
AUTHORIZED	PORTLAND OFFICE:	Address
digital	Don Shafer 9498 S.W. Barbur Blvd.	City
COMPUTER DISTRIBUTOR	Portland, Oregon 97219 503-245-1261	State Zip
	000-240-1201	Telephone
		(Areg) (Number) (Ext.)

SYSTEM MANAGEMENT FOR RSTS/E

By Gary W. Miller, System Software Manager, Garden Way Manufacturing

NOTE: Due to the tremendous amount of code, all programs referenced here are available on the distribution tape.

For the last eight years I have been working on RSTS/E systems, in that time the need for additional programs for system managers has become apparent. Since I have been a system manager for the last four years I feel that I am fairly well qualified to break down those need into areas where the data is available from RSTS/E.

What I have done is to create a group of programs to suppliment or replace some of DEC's CUSPS. With these programs I am able to charge out the system time to users on a consistant basis as well as charge for disk blocks. I am also able to charge for paper usage on either a block or page basis. The following paragraphs will explain each program or modification and how they enable me to give management a 'reasonable' figure to use for charge backs. This software was not intended to give a one-hundred percent figure because the system cost and development time would be unacceptable.

The first program in the package is the ACCGWM.BAS program, which is a replacment for DEC's REACT.BAS. This program allows the system manager to add, delete, and change information for an account on the system. The program keeps the information on the account in a data file for use by other parts of the billing package. In addition to the information that the DEC program requires this program needs an account description and client code be entered. The accounts are stored using logical names instead of physical device names to allow for packs being placed in different drives.

The program BILLER.BAS gathers the system time and number of disk blocks in each account that is in the data file created by ACCGWM.BAS. The program reads thru the data file to get the device and ppn of the account and then uses the read and reset accounting sys call (Prog Man 7-89) to read the accounting information from the system. This information is accumulated in the data file after the system data is zeroed. The program then goes thru the account accumulating disk blocks returned by the disk wild card directory lookup sys call (Prog Man 7-100). The total disks blocks in the account is added to the data file and the sample count in the file is incremented as well. On our systems we run this program every night to allow us to get a 'reasonable' average disk block number.

The MONTH.BAS program reads the data from the disk file created by ACCGWM.BAS and prints it out in either ppn or client code order. This program will also allow the user to zero out the data at the end of a billing period. The user can also request that the program print the passwords on the report if that is required.

The PASWRD.BAS program allows the user to change his/her password at any time without the need to contact the system manager. This also means that the responsibility for account security is now on the user, where it should be. This ability would shake up some shops because there maybe times when someone in Data Processing may need to get into the user account and getting the password could be a problem. In a later section this problem will be addressed as to my solution to the problem.

The next section of this article will deal with modification to the DEC CUSPS that I have made to control use of the system as well as measure what the system is doing.

The first modification that was made was to the LOGIN.BAS program to add special features for ease of system use. There are a number of features that were added so I will just list them with text to explain what they are and why they were made.

- The first change to the program was to force the user to see the \$NOTICE.TXT file, if there is one, all the time. This was done to prevent users who always use '/' on login from missing important notices. I think a better way would have been to print a message that there was a notice on file and maybe even print the first line of the message so if the user had already read the message they need not read it again.
- 2. The next change was to allow the programming department the ability to get into user accounts without knowing the password. On our system all of our programmers are priviledged users, even though this is not the usual case the ablity to move into other account is a requirment of most programming departments. What was done was to create three super passwords with certain abilities to each. There is one that will let you into any non-privileged account, one to let you into any priviledged account other than those connected with system management, and the last one will let you into any account on the system. Now I realize once you have let a programmer into a priviledged account they could write a program to read passwords from the system, but in a reasonable shop this may be something you could live with. With this patch just for non-priviledged accounts you could then live with users controlling their own passwords.
- 3. The next change was build a crude auto-baud feature into the program for dial up lines. This only affects logged out users on dial-ups. When the line first comes active the line is set up to come up at 1200 baud with the '/RING' option of TTYSET since the fastest dial-up here is that speed. The keyboard is then opened binary mode and the user must type CTRL/C, the code then checks the buffer for a CHAR\$(3%) and if found then the speed of the line is correct, otherwise the line speed is changed to 300 baud and the user will have to type CTRL/C again. The same check is made and if the speed is not correct then the line speed is set to 110 baud and then the program continues at the logged out entry. This patch means that the user will have to type CTRL/C at most two times to set the line speed since I only support the three speeds.

- 4. Since use of our dial-up is critical at night LOGIN was changed to prompt for a special password on dial up lines after 4:30 pm. This password is set up to automatically change on a daily basis so even if you give out the password it will only be good for one day.
- 5. There has been a lot of talk about quota enforcement on RSTS/E and how it should be done, but none of the solutions seem to be feasable without massive changes to the system. My solution for what ever it is worth was to enforce quota on login as well as logout. The program standardly enforces logout quota, so all I had to do was add the login quota check. This is set up so that if an account is over quota then no further users will be able to sign into the account. If the user does not sign off and

Since use of our dial-up is critical at night LOGIN was the system is shut down then they will not be able to sign back in. The user would then have to contact someone in D.P. to get there account cleaned up. This has been very effective for us.

The following is a list of additional programs and patches to increase the flexability of the system.

1. ERRDET.BAS

- 1. This program was modified to print the file name and information in the FCB and WCB if both were available. This allows me to find out what file may have a bad block and what the block is.
- 2. BATCH.BAS
 - These programs were modified to only run from the ppn where the programs are compiled. This was done to prevent the spread of works files into full accounts as well as prevent unauthorized personnel from starting batch spoolers.
- 3. QUE.BAS
 - This program was modified to prevent the Queueing of batch streams by nonoperations personnel. We have 24 hour coverage from Sunday midnight to Friday midnight and the operations department controls all the scheduling. To allow for weekend system use the patch allows all users to queue on Saturday and Sunday. This patch does not affect any queues that are done from a batch stream.
- 4. SPLRUN.BAS
 - This program was modified to skip sending the device hung message if nothing has been printed yet. This was a request from our operators and just keeps the OSC from getting extranious messages.
 - This next patch was made to help a system manager charge back for paper

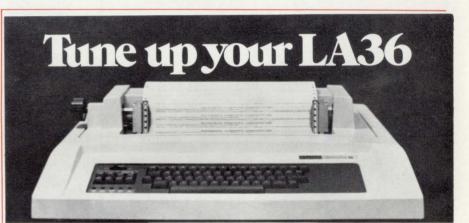
usage. The program now creates a file with the following information in it

- 1. The full file name of the file being printed.
- 2. ppn of the Queuer.
- 3. Form name.

RSTSPROFESSIONAL RSTSPROF

- 4. Date Queued.
- 5. Time Queued.
- 6. No. of Pages (if spooler started with NOFORM)
- 7. No. of Blocks
- 8. Job name

The file created by SPLRUN.BAS is named according to the receiver ID of the spooler and is opened in the account that the spooler is running out of. This file or files, if you have more than one printer, is read and combined into a



The DS120 Terminal Controller makes your LA36 perform like a DECwriter[®] III.

The Datasouth DS120 gives your DECwriter[®] II the high speed printing and versatile performance features of the DECwriter[®] III at only a fraction of the cost. The DS120 is a plug compatible replacement for your LA36 logic board which can be installed in minutes. Standard features include:

- 165 cps bidirectional printing
- Horizontal & Vertical Tabs
- Page Length Selection
- 110-4800 baud operation
- 1000 character print buffer
- X-on, X-off protocol
- Self Test

- RS232 interface
- 20 mA Current Loop interface
- Top of Form
 - Adjustable Margins
 - Double wide characters
 - Parity selection
 - Optional APL character set

Over 5,000 DS120 units are now being used by customers ranging from the Fortune 500 to personal computing enthusiasts. In numerous installations, entire networks of terminals have been upgraded to take advan-

tage of today's higher speed data communications services. LSI microprocessor electronics and strict quality control ensure dependable performance for years to come. When service is required, we will respond promptly and effectively. Best of all, we can deliver immediately through our nationwide network of distributors. Just give us a call for all the details.



data South computer corporation 4740 Dwight Evans Road • Charlotte, North Carolina 28210 • 704/523-8500

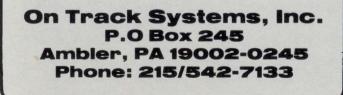
CIRCLE 73 ON READER CARD



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 \$995 single cpu license
- Trial Version \$100

For more information contact:



master file by the program SPLPUF.BAS. How this master file is used is up to the system manager. The recommended procedure would be have all queues be done with a job name describing what department the report is for and what form the report is being printed on. In our case we use the first three characters of the job name for the department and the last three for the form being used. This allows the master file to be sorted by job name/file name and then a report printed breaking on change in file name then breaking on change in form type and then breaking on department. This report can then be broken apart and then sent to the respective departments to show their usage.

The last program that will discuss is the program we use for our weekly backup disk to disk. Our policy is to use SAVRES on Tuesday, Wednesday, and Thursday nights to backup the system. All of these backups are done in IMAGE mode disk to disk. Since the directories on RSTS/E system seem to get very disorganized in a short period of time, the ability to rebuild the directories on a regular basis is beneficial. The program is set up to run on systems with at least three drives of similar capascity. The operator is prompted for the disk being backed up and the drive to write to. The destination drive is then locked and an attempt is made to dismount the pack in the drive, any failures are reported to the operator with appropriate corrective measures. To insure that the wrong pack is not mounted the program requires the the pack to be written to has the same pack ID as the source pack. When the correct pack is mounted then the program opens a PK and starts a dialog to copy the source disk to the output disk. The dialog starts by running DSKINT.BAS to clear the pack of information and then using UTILTY to mount the pack. After the pack is mounted the program MAKMFD.BAS is run to create a MFD on the output pack with the accounts in ascending order. The copy is then done by using PIP.SAV running at 28K to speed the transfer of data. When the copy is complete the pack is dismounted and the operator is requested to put the original pack back in the output drive. This proceedure is not the best method for backup but the UFD's seem to be in good shape after the proceedure. With this program we can copy four 300 mb disk drives in just under three hours.

These changes have been made and the programs written to satisfy the demands of managment, but they also serve the D.P. department and the user. I hope that this kind of information is of interest to you the reader.



LETTERS to the RSTS Pro continued from page 6

Since we had the older version 11/34 (as opposed to 11/34A) we were informed that DEC cache was not compatible. Hence we purchased an ABLE CACHE/434 and installed it in late 1978. Installation was a simple matter of removing the UNIBUS jumper between two backplane sections and replacing it with the ABLE CACHE/434 which consequently occupied no backplane slots.

A word of warning is appropriate here - the only memory that wil be cached must physically lie between the CPU and the ABLE CACH/434. If enough backplane slots are available to permit all memory to be installed between these two components, there is no problem. Otherwise, the cache will not be used to its fullest capacity.

We did not perform extensive bench marking, however we observed substantial improvements in certain operations. For example, compiles and task builds of BASIC PLUS 2 programs were perceived to decrease from 7 CPU minutes to 4.5 CPU minutes. We recognize that this is not an especially significant example since task building may not be representative of general system operation. However, since we have been plagued with long task build times, this improvement was viewed with some enthusiasm.

General system response was observed to improve somewhat (a very subjective statement) but no quantitative measures were taken. In retrospect, a careful analysis of STATUS reports before and after should have been performed.

I am not especially optimistic about the cost effectiveness of adding cache to an 11/34A using RSTS since most systems of this type are disk bound due to heavy swapping requirements rather than CPU bound.

A careful analysis of STATUS or STATS reports or VT50PY during heavy usage periods will probably show high "lost time" figures. These figures will not be substantially improved by adding cache. It is possible to relate the lost time figures to the amount of swapping being performed.

This may be accomplished by determining the number of swaps over a period (say 60 seconds) and the number of blocks swapped. The total time used for swapping will be:

T = # swaps x access time + # blocks x time to transfer 1 block

There are only two ways to reduce this time T:

1) to add more memory (not often possible on an 11/34A)

2) use a faster swapping medium.

Solution 2 can be explored if there are faster disks available than those currently being used for swap files.

We investigated the possibility of using the so called "solid state disks" but rejected that approach as not cost effective for the following reasons:

1) The data transfer time represents a significant portion of the swapping time, T (particularly for systems with BASIC PLUS 2 jobs, ie. larger than 16 K swap max).

A glance at the specifications for the faster disks (RP06's and RM02's) will show that their transfer rate is similar to that of a "solid state swapping disk".

2) Although the access time for a "solid state swapping disk" is substantially less than for an RP06 or RM02 (1 micro-second vs. 50 millisecond) this does not have a large effect on the overall swapping time T.

The "solid state swapping disc", however, would be an excellent choice for heavily used read only files (such as tables), or heavily used temporary files (such as work files). 3) Swapping time T will be somewhat smaller for a "solid state swapping disk" but the high cost of this type of storage will generally preclude the possibility of placing all swapping files on such a device.

We were able to test this theory earlier this year when we added RP04 drives to our system and moved our swapping files from RP03 drives to the RP04 drives. The transfer rate improvement of a factor of three resulted in lost time dropping to about a third of our original lost time.

If Mr. Horst's system is already using fast disk drives, he should try all the system optimization procedures available and as outlined in a recent *RSTS Professional* article by Dave Mallery, entitled "A RSTS Performance Checklist", V.2, #4.

In summary, a PDP 11/34A using RSTS with twenty-five jobs is probably swap bound and would benefit more from faster swapping facilities than from cache memory. If no faster swapping facilities are available, I believe Mr. Horst would be better served by saving his money until he can purchase an 11/44 with more memory. We have just done this and our response improvement is substantial.

I would be pleased to make my analysis of "solid state swapping disks" versus conventional disk drive available if any one is interested.

We are delighted with the *RSTS Professional* magazine and look forward to each new issue.

Yours truly, David J. Leffen, Ph.D., P.Eng. Vice-President

G.K. Fleming & Associates Ltd.

Thunder Bay, Ontario

Thank you for your thoughtful response. We are looking forward to publishing the article (your "analysis") in a future issue.

Dear Editors and Staff:

Congratulations on your anniversary! Please accept this token of appreciation from all of us here at Software Techniques.

You all have something to be very proud of. You have taken an idea and turned it into a reality, building something of value where there once was nothing at all.

It has been a pleasure to work with you over the last year. May you have many more happy anniversaries.

Sincerely, Rick

And all of us at Software Techniques Los Alamitos, Calif.

Thank you, Rick and staff. With friends like you we're bound to succeed.

Dear RSTS Semi-Professional,

Tri-State is not renewing its subscription to your magazine. We have a VAX now, and can't wait til the 11/70 goes out the door. Maybe it will find a nice home in the back of a Mom-and-Pop Drug Store or somewhere like that.

Whenever I think of your magazine, I see in my mind that editorial on page 4 of the Feb/March 1980 issue. Mr. Mallery can keep his little RSTS machine forever, I don't care. But we are happy to leave it for good.

In 20 years, when most of the world has progressed even further, I'll think of you, soldering your own boards, keeping those 5 or 6 users happy. Sincerely, Phil Jamieson Data Processing Manager

Tri-State G&T

Good Luck with your VAX. We think you'll be interested in the VAX-SCENE, published, of course, six times a year in the "RSTS PRO." My good friend David tells me of a lovely way to soft crash any RSTS machine ...

All you have to do is to poke the address of your clock interrupt routine, and this will cause a crash the next time the clock ticks (ie, within a 50th/60th of a second).

- Hello 1.1
 - (Password)
 - Switch BASIC

w\$ = SYS(CHR\$(6%)+CHR\$(-6%)

+CHR\$(64%)+CHR\$(SWAP%(64%)) Please note that location 64 (decimal) is the address for a LINE clock, and that anyone with a PROGRAMMABLE clock should poke at location 68 (decimal). Bye Y, Peter Dick The enclosed is from the September 1981 New-

sletter of the DECUS RSX UK SIG.

I think it deserves a larger readership.

It is with great pleasure that we announce a new product — FINS-11. This product is a special hardware/software integration package developed by DEC and Tidewater Research Systems to produce a militarized version of the PDP-11 for use by the U.S. Navy. It is fully submersible and can travel at 15 knots due to some recent hardware breakthroughs we have made which resulted in the TUNABUS. FINS-11 also features the OCTOBUS which is a high speed bus with 8 ports.

During highly rigorous and lengthy field tests, during which time FINS-11 terrorized twelve towns of the Rhode Island coast and at Cape Cod last summer, the Underwater Research labs used FINS-11 to do process control on clam beds using special commuications software called DECwet-11. The Navy particularly liked the FINS-11 D/A converter ROM package which plays "Anchors Aweigh" whenever it comes out of the water (Richard Rogers' "Victory at Sea" is also available but requires the special bouyancy option for the ROM).

Field Service has been attending special classes for SCUBA training off Martha's Vineyard to enable them to tackle support of FINS-11. FINS-11 is supported in HOSS by Martin Minnow.

- Special hardware features of FINS-11 include:
- High powered cooling fins
- Bubble memory
- SQUID ink injection line printer
- Micro-fish reader (GUPPY)

 Admiral mode replaces Kernel and Supervisor mode. Software features of FINS-11 include:

• A very forgiving operating system — COS-11 (Chicken of the Sea), written in CORAL with realtime extensions

- Mackeral-11 assembler
- FILET file transfer system
- A powerful user interface, COD (Command Operations Decoder)
- An Unlimited freshwater node pool

• STARFISH - VAX emulator

• The block structured ALGAE-60 high level language

• CRABS (Comprehensive Random Access Batch System), which allows jobs to be spawned up to the batch stream

• The heart of COS-11 is the ATOLL scan which will trap tasks that are floundering or wading for nodes.

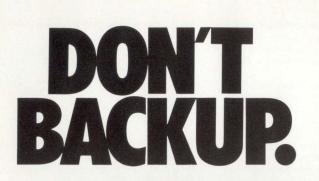
Autopatch is accomplished by running SUBS (Systems for Updating Binary Software) in amphibious mode.

One of our European field test sites was Jacques Cousteau, who ran a special version of the system, called OYSTER, which was written in PEARL. Since he is such a distinguished user, we substituted John Denver's "Hail Calypso" for "Anchors Aweigh".

Yours as usual, Peter Dick, Prop. Silver Programs, London, England

Beautiful, Peter. Thank you.

AT LASTIII The RSTS PROFESSIONAL will be published six times a year. February, April, June, August. October, December



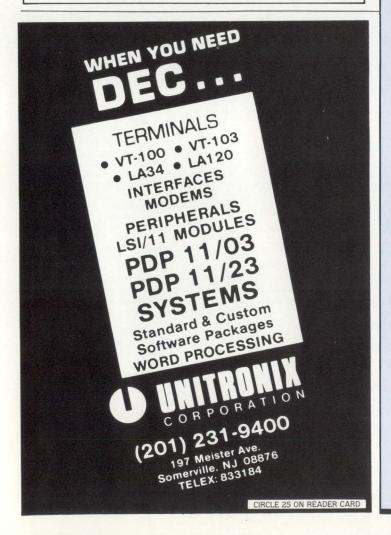
If your system backup puts everything else on the back burner, you need SAVER for RSTS/E.™ It's the only disc to tape backup that doesn't monopolize your PDP-11[™] system resources.

SAVER restores files easily, and creates optimallystructured discs. And it's supported by Data Processing Design, developers of WORD-11 and other fine products for DEC's family of mini-computers. Inexpensive at \$850.00. Call or write.



Corporate office: 181 W. Orangethorpe, Suite F., Placentia, CA 92670 (714) 993-4160 New York Office: (212) 687-0104 Washington, D.C. Office: (301) 657-4098

CIRCLE 74 ON READER CARD



BENCHMARK DIBOL VS. BASIC+2 PDP/11-RSTS/E Environment

Analysis Conducted by Frank Metcalf, ICS Computing, Ltd., UK

Benchmark Overview

The following benchmark results show advantages for each language (BP2 and DIBOL). However, the more important advantages appear to be in favor of BP2.

Two of the more important considerations in a RSTS/E environment are disc accessing and CPU usage. In each of these areas, BP2 was clearly the most efficient. Also, job mix is of major importance in a RSTS/E environment. BP2 is clearly the best performing language.

The above points should be key considerations in evaluating languages and operating systems in any environment where development and production are on-going.

The general feeling, prior to the comparison, was that BP2 was a better language for the RSTS/E environment, which the benchmark has proven.

The attached should provide sufficient detail to support the findings.

Comparison of DIBOL vs. BASIC+2

System compared: Supplier masterfile maintenance 1300 records on VENMAS.

1. COMPARISON OF DISC ACCESSING

The figures represent the disc accessing counts for the data and index files only. Directory, DEVICE.DDF and overlay accesses are ignored.

Each activity was performed 5 times.

		Rea	ds	Wri	tes	Total
Activi	Activity		Blocks	Accesses	Blocks	Accesses
Search fails	BP2	12	22			12
	DIBOL	157	157	2	2	159
Inserts	BP2	28	54	5	10	33
	DIBOL	240	240	18	18	258
Amends	BP2	17	32	5	10	22
	DIBOL	192	192	8	8	200
Deletes	BP2	17	32	5	10	22
	DIBOL	202	202	18	18	220

For the DIBOL systems, no count was taken of the accesses required to sort the index to include recent inserts or to remove deleted records from the data file.

The amount of disc usage has an important effect on machine performance. Thus, the above results point very much to BP2.

2. COMPARISON OF CPU USAGE

Each activity was performed 10 times.

Activity	BP2	DIBOL
Inserts	6.6	21.0
Amends	6.4	10.5
Deletes	6.0	7.0

For the DIBOL system, no count was taken of the CPU time required to sort inserted records into the index or to remove deleted records from the file.

This also points to BP2 and, as with 1, is an important factor on the 11/70's.

3. COMPARISON OF TSK FILE SIZES

Basic Plus 2 s	systems:			
VENMNT.	-	161	(176)	
DIBOL system	ns:			
VENMNT.	TSK	25	(32)	
VENPRT.	TSK	23	(32)	
ORGVEN.	TSK	22	(32)	
VENCNT.	TSK	7	(16)	
SRTVID.	TSK	27	(32)	
111			<u></u>	
Total		104	(144)	

The figures, in brackets, represent the total blocks allocated, allowing for a disc clustersize of 16.

The file size above for BP2 will not decrease significantly for even very small programs. However, only 1 file is required for a module, whereas, 5 programs make up a DIBOL module. Disc space on 11/70 is not a problem, although the size of a directory could be. There is probably little advantage either way. Also this has a less important impact on performance than either section 1 or 2.

4. COMPARISON OF FILE SIZES

RMS uses more space to hold indexes and allow for inserts (depends on packing density).

RMS packs numeric fields more efficiently (in binary).

In the systems used for comparison, these 2 factors balanced to give equal file sizes.

However, the number of files in RMS system would be smaller since indexes and data files are incorporated in 1 file, unlike DIBOL.

5. COMPARISON OF TASK BUILD TIMES

Basic Plus 2:

1 min., 2.9 secs.	CPU time
18 mins.	Elapsed time

to build VENMNT.TSK

DIBOL:

14	4	secs.	
3	n	nins.	

CPU time Elapsed time

e

to build VENMNT.TSK

6. MISCELLANEOUS

DIBOL programs are scheduled and run for each character entered. This produces (1) a heavy system load and (2) delays in echoing characters entered, which is annoying to a user.

BP2 is no more difficult to use than DIBOL, although it could take a little longer to learn.

BP2/V1.6 and V7.0 of RSTS/E have improved Task (TSK) file sizes, compilation time and overall performance. Benchmark times were not run against BP2/V1.6 and RSTS/E V7.0.

7. CONCLUSION

BP2 is clearly more suitable to a RSTS/E machine than DIBOL. The figures in numbers 1 and 2 indicate considerably more BP2 jobs could be run on the 11/70 than DIBOL.

Stocking genuine **DEC** computer spares

Spare Parts for: **DEC PDP Computer Family** DEC Line Printers DEC Video Screens LSI-11 Others

Call Radgo. We handle genuine DEC parts manu-Factured by DIGITAL EQUIPMENT CORPORTION. We sell at factory prices with FACTORY WARRAN-TY. Radgo is stocked and staffed to help you with your requirements for most general purpose K & M series and many COMPUTER SPARE MODULES and COMPONENTS. We also stock a large assort-ment of COMPUTER SUPPLIES.



To order or for free catalog call 1-800-543-1986. Ohio customers phone 1-513-752-6880.

3988 McMANN RD., CINCINNATI, OH 45245 CIRCLE 75 ON READER CARD

Hardware Accessories For DEC Equipment Users

C-XX Overtemperature Protection System

Standard DEC PDP 11, VAX, and System 10-20 machines are NOT adequately protected from equipment damage due to high machine room temperatures. This unit provides aural warning signal and total system power shutdown with two customer adjustable temperature limits and approved interface to standard DEC AC power control system.

B-11 CPU Speedup for PDP 11/45 and 70

This timing generator modification allows 11/45 or 11/70 CPU instruction throughput rate to be increased by up to 15%.

Both products are attractively priced, install in a few minutes with no special tools or skills, and are allowed by DEC Field Service in contract systems. Call or write for further information.



Nassau Systems P.O. Box 19329 Cincinnati, Ohio 45219 (513) 231-1283

DEC, VAX, and PDP are trademarks of Digital Equipment Corp.

OVER THE I/O PAGE

(or, 'A Trip Through Magic Kingdom . . .')

By Bob "MACRO MAN" Meyer

The following article describes in detail a method of accessing the I/O page of the PDP-11 while running under RSTS from a user-mode Macro program. Also we'll cover some possible uses for this type of programming, as well as some potential dangers.

This project came into existence a few years ago when a good friend of mine (he wishes to remain nameless, so we'll call him by his alias, 'Farquart') and myself decided how nice it would be if we could 'peek' into XBUF. We realized very soon that this was not at all feasible using existing techniques. "But, Farquart," I exclaimed, "RSTS certainly has access to the XBUF area, why can't we?". In other words, what is the difference between the Monitor accessing memory, and the user program accessing memory?

The answer: Memory Mapping.

In order to look around the system's memory at will, the user program must have the ability to re-map itself as needed: And in order to re-map one's own job image, we must have access to the Memory Management APR's. The following examples will demonstrate how this can be done on any RSTS machine. I would like to recommend, however, that you read this article throughly before trying any of this, and if you're not intimately familiar with the PDP-11 Memory Management hardware, do your experimenting with a stand-alone system.

Building the Library

The first step is to build a resident library (reslibs must be genned in, of course) with a length of 4K. This library will then be placed at the very top of the system's memory, and with a gentle 'shove', will land directly over the I/O page. The program below will do just fine:

.TTTLE USR07 .IDENT /1.0/ USR07: .BLKW 4096. :BLOCK OUT 4K WORDS .END : to assemble:

MAC MAC>USR07 = USR07 MAC>1Z

now to link up the reslib:

TKB TKB > USR07/-HD = USR07 TKB > / ENTER OPTIONS: TKB > PAR = USR07:160000.20000 TKB > STACK = 0 TKB > //

TKB>PAR=USR07:160000:20000 TKB>TRS-PAR=USR07:160000:20000 :STARTING ADDR & LENGTH OF LIBRARY :DON'T ALLOCATE ANY STACK SPACE

and finally, we create the library image file:

RUN \$MAKSIL [maksil's header] Resident Library name? USR07 Task-build Resident Library input file < USR07.TSK>? < CR> Include symbol table (yes/no) < Yes>? NO Resident Library output file < USR07.LIB>? < CR> USR07 build in 4 K-words. 0 symbols in the directory USR07.TSK renamed to USR07.TSK < 40> We now have a USR07.LIB file in the current account. To put up this library we type:

UT ADD LIBRARY [p,pn]USR07>40>/ADDR:xxx/STAY/RW

where:

[p,pn]is the current account, and xxx is the amount of memory on your system minus four so if your system has 512K Words, specify '/ADDR:508'.

Finally, we convince RSTS to let our library live over the physical I/O page of the system. First we log into [1,1] and run ODT:

RUN \$ODT [odt's banner] File < Memory >? < LF>

Type the letter 'C' to get a listing of all monitor tables, watching near the bottom for 'RTSLST'.

. JOBCNT 072564 RTSLST 056306

(note: these values are very system dependant)

The list of Resident Libraries, 'LIBLST' starts just after 'RTSLST'. So in order to access the first entry in the RESLIB list on this system, we would look at location 56310:

*56310/ 006540

However, we're interested in the last library in the list (the one we just added:USR07), so we must traverse the library links as follows (in this case, we have two other libraries before USR07):

*56310/006540 @ 006540/005200 @ 005200/015600 @ 015600/000000

We're now at the start of the memory control subblock for the third library on the system. (see common.mac for more details.) To verify this, we can check the name of the library:

> 015600/ 000000 < LF> 015602/ 103112 %USR ;use '%' to get rad50 string 015604/ 140512 %07

If you get the library name in the second two locations of the control block, you're in the right place. If not, back up and start over.

Now that we've found the entry for our library, we can change it's physical start address so that the Monitor 'thinks' it exists over the I/O page:

15620/ 056400 < i/o page address > < CR >

where $\langle i/o \rangle$ page address \rangle is 177600 for the 11/70 & 11/44, and 7600 for all others.

Accessing the Library

A simple way of trying out this new Resident Library is to link RSXODT with it, and nose around a bit: (you should be somewhat familar with ODT at this point) First

HAVE WE GOT A MESSAGE FOR YOU!

INTECOM brings to RSTS users an extremely versatile message system. Much more than 'electronic mail,' **INTECOM** is an application package designed expressly for controlled user communications, correspondence management, and item scheduling.

INTECOM can be used for short, paperless office memos; for lengthy, printed reports; or for any text file exchanges. **INTECOM** interfaces with the editor of your choice for complete text formatting capabilities.

A built-in calendar function permits you to date messages for automatic future transmission to any valid **INTECOM** user -- including yourself. Incoming messages may be placed on hold for specific periods of time. Communications on hold disappear temporarily from your 'in-basket' and re-appear on the scheduled reactivation date! **INTECOM** even notifies the user's designated terminal when incoming messages arrive. **INTECOM**'s unique message labelling feature permits you to store your own key identifiers for message retrieval. You can identify messages by keyword, sender, receiver, date, time, or subject.

INtelligent TExt COMmunicator Specifications

- user name logicals
- user/account validation
- menu operation with full HELP text
- IN, OUT, and HOLD transaction databases for each user
- message reply routing
- text file or text record manipulation
- hardcopy queueing (message selectable)
- message labelling for keyword retrieval (incoming and outgoing)
- full message journalling (user selectable)
- message waiting light for VT100 (user selectable)

- bell notification of message receipt (user selectable)
- incoming message forwarding with supplementary text
- message sending to multiple users
- message serialization, date and time marking
- dated message sending (relative days or absolute date)
- dated message suspensing (relative days or absolute date)
- multiple account operation for each user
- text editor interface from menu control (user selectable)
- message subject scanning with selective read
- incoming, outgoing, and dated message reporting

INTECOM's features can be applied to every application environment. Contact us for the impressive details. **INTECOM** is being offered at an introductory price \$2880. Demonstration plans are available.

INtelligent TExt COMmunicator ...

another QUALITY software product from:



North County Computer Services, Inc.

2235 Meyers Ave., Escondido, California 92025 (714) 745-6006, Telex: 182773

RSTS is a registered trademark of Digital Equipment Corporation. © 1981 Logic eXtension Resources.

DEC SYSTEM SUPPORT

RESOURCE-11 offer's complete system support for OEMs and end users of Digital Equipment Corporation (DEC) systems. Professional evaluation of your needs before and after your investment. RESOURCE-11's technical expertise encompasses the entire PDP-11 and VAX/VMS product line.

DISK SAVE & COMPRESS (DSC)

DSC permits pre-extension of RSTS/E UFDs, data compression, and manual placement of files for disk seek optimization on a total system scale.

WORD PROCESSING

Fully supported packages under RSTS/E, RSX and VMS available for large, small and standalone systems. Easy preparation of a wide variety of documents such as letters, reports, contracts and technical material with ease.

SYSTEM EVALUATION & OPTIMIZATION

Step by step evaluation of future and existing systems. Let RESOURCE-11 maximize your hardware and/or software investment for maximum performance and usability.

For complete details, contact us at:

RESOURCE-11

19841 Cochrane Way Gaithersburg, MD 20789 (301) 258-9606

Don't buy a VAX . . . Call MACRO MAN

for superior performance **RSTS** internals custom Macro programming **RSTS or 11/M**

SPECIAL OFFER

LEARN MACRO FROM THE MASTER! One week course in Philadelphia call for details.

MACRO UTILITY LIBRARY build your own! - call for information

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

create a small macro program consisting of nothing but a Breakpoint instruction:

CIRCLE 77 ON READER CARD

TITLE JUNK .IDENT /JUNK/ IUNK. BPT END JUNK

and assemble: MAC JUNK = JUNK

and link with the resident library:

TKB TKB>JUNK/DA=JUNK TKB>/ ENTER OPTIONS: TKB>RESLIB=USR07/R0 TKB>//

SPECIFY READ-ONLY ACCESS FOR NOW

;/DA CALLS IN RSXODT

and we're ready to run:

RUN JUNK ODT-JUNK

177560

177562.

177564

177566

If all was done correctly, examining any address from 160000 thru 177776 should return data from the I/O page. If you are familar with any particular device and its operation, you may wish to examine its registers to verify that this really works. One simple test is to use the console DL CSR's:

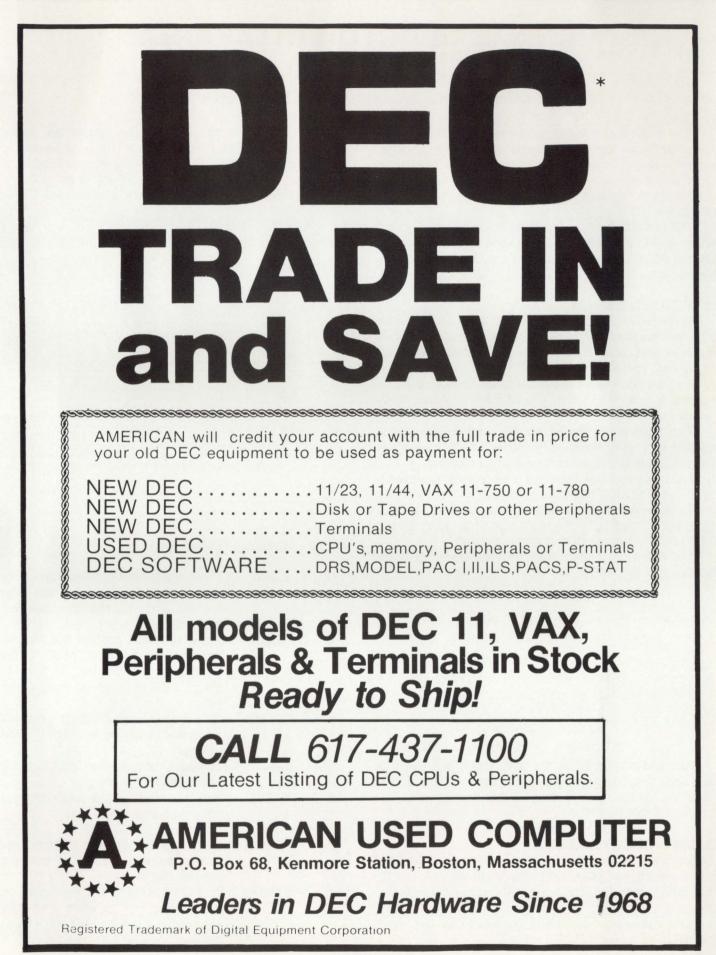
/ 000100	RECIVER 'INTERRUPT ENABLE' BIT ON
/ 000215	:LAST CHARACTER RECEIVED (< CR> IN THIS CASE)
/ 000200	TRANSMITTER 'DONE' BIT ON
/ 000000	:XMIT DATA REGISTER (ALWAYS ZERO WHEN READ)

And if ODT was task-built with the '/RW' switch on the RE-SLIB option, you can actually write a character out to the device:

177566/ 000000 101 < CR>

This will write an upper-case 'A' to the console terminal. That's all for now; be careful and enjoy.

(P.S.: XBUF Display will go on sale soon! Contact Macro Man for details.)



RSTS/E MONITOR INTERNALS Part 4

By Mike Mayfield, Northwest Digital Software, Box 2-743, Newport, WA 99156

This is the last article in my series on RSTS/E monitor tables. By this time you should be getting a better understanding of how RSTS works inside and where to find the information you need for specialized systems programs and system tuning. Now all of the monitor's secrets are just a PEEK away.

This last article covers the data structures used by send receive and CCL commands. These data structures are the RIB and PMB for send receive and the CCL block for CCL commands.

5.0 SEND RECEIVE

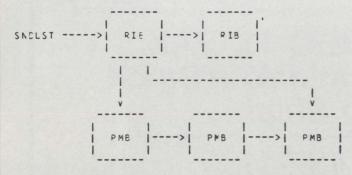
The send/receive capability of RSTS allows information to be sent between programs. The data structures required to provide this capability are the receiver ID block (RIB) and the pending message block (PMB).

Each program that has declared itself to be a receiver is allocated a receiver ID block. This ID block is pointed to by the job's second JDB.

The receiver ID blocks on the system are linked together in a list of message receivers. When a message is sent this list is searched for the specified receiver.

Each message that is waiting to be received has a pending message block (PMB) associated with it. The PMBs for all of a particular receiver's pending messages are kept in a linked list pointed to by the word S.MLST in the receiver's RIB. The list is kept in first-in first-out order.

The following figure shows the relationship between the RIBs and the PMBs:



For more information on the contents of the receiver ID block and the pending message blocks see the "RSTS/E Programming Manual" and "Network Programming in BASIC-PLUS and BASIC-PLUS-2".

5.1 RIB - RECEIVER ID BLOCK

When a job declares itself a receiver (using the .MESAG monitor directive or SYS(6 + 22) in BASIC-PLUS) a receiver ID block (RIB) is allocated. The receiver ID block contains all the information necessary to allow receipt of intra-CPU and inter-CPU messages. The RIB is pointed to by the word J2MPTR in its owner's JDB2.

In addition, all the RIBs in the system are linked together in a list. The first element in this list is pointed to by the location SNDLST. Since the receiver name ERRLOG is always present in the system, SNDLST will never be 0.

The format of a receiver ID block is as follows:

	Link to next RIB	C	S.LINK
		2	S.RCIC
	Receiver ID	4	
	(ć bytes of ASCII)	ć	
S.OBJT 9	Object type Job number *2	8	S.JBNC
	Unused Access flags	10	S.ACCS
	Buffer maximum	12	S.BMAX
S.MCNT 15	Pending count Message maximum	14	S.MMAX
	Pointer to first PMB, if any	16	S.MLST
	Pcinter to last PME, if any	18	
S.LCNT	Link count Max link count	20	S.LMAX
	Fointer to first logical link block	22	S.LLST
	Pointer to last logical link block	24	
		26	
	Reserved for network use	28	
	ll	30	

offset	Symbol	Description
0	S.LINK	This word contains the address of the next RIB in the linked list. If this is the last RIB in the list this word will contain a 0.
2	S.RCID	These six bytes contain the receiver name as six bytes of ASCII. If the receiver name is less than six bytes in length it will be filled with trailing zero bytes.
8	S.JBNO	This byte contains the job number times two of the job to which this RIB belongs.
9	S.OBJT	This byte contains the object type for use in DECNET messages.
10	S.ACCS	This byte contains the access controls for this receiver (see 5.1.1).
1	S.BMAX	This word contains the buffer usage restrictions specified at the time the receiver was declared.
4	S.MMAX	This byte contains the restriction on the number of pending messages specified at the time the receiver was

declared.

page 89



SYSTEM DEVELOPMENT TOOL FOR RSTS/F BASIC -PLUS

- Automated Definition of Record Lavouts
- ISAM, Sequential, Random, and Tape Files
- Version 7 RSTS/E Large File support
- Program generated, RNO compatible Document Facility
- Supports numerous Data Types
- DMS 500/FAM Compatible

CASHER ASSOCIATES INC.

1371 Beacon Street · Brookline, MA 02146

(617) 232-9111 (212) 757-2868

CIRCLE 78 ON READER CARD

15	S.MCNT	This byte contains a count of the number of messages that are currently pending for this receiver.
16	S.MLST	This word is a pointer to the pending message block (PMB) for the first mes- sage pending for this receiver. If no mes-
18		sages are pending this word contains a 0. This word is a pointer to the pending message block (PMB) for the last message pending for this receiver. If no message are pending this word points to S.MLST.
20	S.LMAX	This byte contains the restriction on the number of pending DECNET messages specified at the time the receiver was declared.
21	S.LCNT	This byte contains a count of the number of logical links for DECNET messages
22	S.LLST	currently pending for this receiver. This word is a pointer to the logical link descriptor block for the first logical link pending for this receiver. If no logical links are pending this word contains a 0.
24		This word is a pointer to the logical link descriptor block for the last logical link pending for this receiver. If no logical links are pending this word points to S.LLST.
26-31		Reserved for network use.

5.1.1 S.ACCS — Access Control Bits

The access control bits contained in S.ACCS have the following meaning (see the RSTS/E Programming Manual for more information on access control bits):

Bit Symbol Description

0 SA.LCL Local senders may send to this receiver.

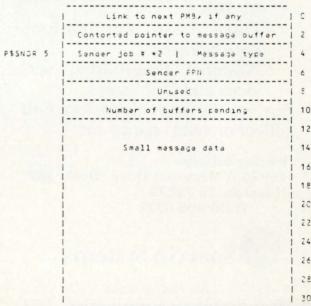
1	SA.PRV	Local senders must be priviliged to send to this receiver.
2	SA.NET	Network senders may send to this receiver.
3	SA.1SH	Network senders may only send if no logical links are already pending (i.e. single message mode).
4-6		Unused.
7	SA.XOF	Receipt of messages from local senders is temporarily disallowed.

5.2 PMB — PENDING MESSAGE BLOCK

Each message waiting to be received has a pending message block (PMB) associated with it. The PMB describes the data received for large messages and contains the parameters sent for small messages.

The pending message blocks for each receiver are kept in a linked list ordered first-in first-out. The first and last pending message blocks in the list are pointed to by the two words starting at S.MLST in the RIB for this job (see 5.1).

The format of the PMB is as follows:



Structured disks for all! Announcing REACT2.TSK

- a 'REACT' replacement that locates & extends UFD's
- all standard 'REACT' functions (Delete, Standard, Enter)
- user specified location and length for new UFD.
- high speed this product uses software developed and licensed by Software Techniques, Inc.
- distributed as an RSX or BP2 task on 9 track 800/1600 tape

Introductory Price: \$200.00 **Single CPU License**

Nationwide Data Dialog

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

Call For Ouick Service CIRCLE 7 ON READER CARD

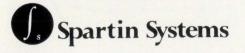
DEC/RSTS

Rapidly growing Houston, Texas OEM seeking RSTS Programmer/ Analysts

- Suburban location
- · Large data center with five 11/70's
- Data base applications development
- · National timesharing services and OEM sales

For quick consideration call collect or send resume to:

Murray Stinson 14925-A Memorial Drive, Dept. 389 Houston, Tx 77079 (713) 496-0771



CIRCLE 79 ON READER CARD

AL RSTSPROFESSIONA	LESTSPROFESSIONALE	STSPROFESSIONALRST	SPROFESSIONALRSTSP	ROFESSIONALRST

Offset	Symbol	Description
0	P\$LINK	This word contains the address of the next PMB in the linked list. If this is the last PMB in the list it will cotain a 0.
2	P\$BUFA	This word contains the "contorted" ad- dress of the buffer containing the large message data. If the low order five bits of the address are zero the pointer is to a small buffer. Otherwise, the address has been rotated left seven bits and points into the extend- ed buffer area, XBUF. If this word is 0 no large message data was sent with this message.
4	P\$TYPE	This byte specifies the type of message described by the PMB (see 5.2.1).
5	P\$SNDR	This byte contains the job number times 2 of the message sender.
6	P\$SPPN	This word contains the PPN of the message sender.
8		This word is unused.
10	P\$BREM	This word specifies the number of bytes remaining in the message buf- fer. It can be less than the original length of the message if the message was too long to fit in the user's buffer on a receive request and truncation was not requested. See 5.2.2 for infor- mation on the buffer format.
12	P\$PARM	These ten words contain the user specified parameters for small

message send/receive and the network parameters for DECNET.

5.2.1 P\$TYPE — Message Type

If DECNET is not used, the message type (P\$TYPE) will always be -1 to signify local sender message received. However, if DECNET is used several other message types are possible, as follows:

- Type Description
- Local sender message received. -1
- -2 Connect initiate received.
- -3 Connect confirm received.
- -4 Connect reject received.
- -5 Network sender message received.
- Interrupt received. -6
- -7 Link service received.
- -8 Disconnect received.
- -9 Link abort received.

5.2.2 Buffer Format

Each large message is stored in a buffer of up to 516 bytes using either a block of small buffers or a buffer from XBUF. The actual data of the message is preceded by a two word header, as follows:

Offset	Symbol	Description
0		This word contains the buffer size, in
		bytes.

BSTS SITE MANAGEMENT AND APPLICATION **DEVELOPMENT TOOLS**

14

24

26

APC

an automatic password changer that creates meaningful sixcharacter passwords and updates the ACCT.SYS file, allows selective changing of passwords and produces three informative reports.

□ ENCRYPTION ROUTINES a site security feature which encodes ASCII characters and can be incorporated into any application where sensitive data is processed. Also exists as a stand alone program for encoding and decoding entire files.

M/APS

a menu/authorization processor and application security system that controls user access to menus and applications programs. Uses DEC's VT series CRTs.

□ STANDARD SUBROUTINE IIBRARY

callable macro-11 routines that perform screen and terminal I/O, cursor positioning and many other necessary program functions, including data conversions

SOURCE/FILE CROSS-REFERENCE (XREF) XREF provides cross-reference listings which detail the relationship between source files callable routines, data files and task images.

VT100 ACCOUNTING CALCULATOR a multi-function calculator designed for user's of DEC's VT100 CRTs. Options and features beyond the capabilities of the normal Accountant's calculator.

McHUGH, FREEMAN & ASSOC., INC. 1135 Legion Drive Elm Grove, Wisconsin 53122

(414) 784-8250

(CIRCL	E	57	ON	READER	CARD

wildcard extension (ie. LOGIN.*). These ten bytes contain the CCL command text. The text is terminated by a byte containing 255. This word contains the device name (in ASCII) of the disk device that contains the program to run in response to this

CCL command. A O specifies the public disk structure, SY:. This word contains the device unit number and unit number "real" flag. See the discription of the FIRQB in the RSTS/E System Directives Manual for more information on device naming.

response to this CCL command. A -1 in this word signifies a

28 This word contains a copy of the pointer stored in offset 4. 30 This word contains the parameter word to pass to the runtime system when the program is run. See the .RUN monitor directive for more information on the parameter word.

LAST CHANCE: An updated wall chart showing the linkages between all tables in the RSTS/E monitor is available for \$2.00. Send your requests to Mike Mayfield, Northwest Digital Software, Box 2-43, Newport, WA 99156

A RSTS INTERNALS MANUAL

BY MIKE MAYFIELD

More than 200 pages in a binder.

Published by The RSTS PROFESSIONAL Pre-publication price offer of \$75 extended to February 1, 1982

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

Request and You Shall Receive Answers. On Line. Software for Manufacturers

MS-11 Manufacturing

- □ Inventory Control
- □ Purchasing
- □ Bills of Material
- □ Shop Routing
- □ Work Order Status
- □ Manufacturing Cost
- □ Material Req. Planning
- □ Capacity Planning

System Highlights

- On Line, Interactive
- Fully Integrated
- Easy to Install
- DEC PDP-11 and VAX



FS-11 Financial

- □ Order Entry/Invoicing
- □ Sales Analysis
- □ Accounts Receivable
- □ Accounts Payable
- General Ledger
- □ Fixed Assets

Company Highlights

- 130+ Installations
- Regional Branch Offices
- Installation Support
- User Group
- Call or write today

388 Oakmead Pkwy. Sunnyvale, Ca. 94086 (408) 245-7990 TWX#9103399258

2

4

RSTSPROFESSIONALRSTSPROFESSIONA

This word contains the offset from the beginning of the buffer (including this header) to the first byte to be returned for the message data. It is initially set to 4 and is updated if a receive request overflows the user's buffer and truncation is not requested.

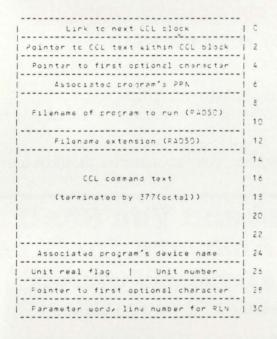
This word starts the message data area. It can be up to 512 bytes in length.

6.0 CCL — CONCISE COMMAND LANGUAGE BLOCK

Each concise command language (CCL) definition is stored in a CCL block. These blocks are allocated and deallocated dynamically from the small buffer pool.

The CCL blocks are kept in a linked list in the order in which they were defined. The first element in the list is pointed to by the location CCLLST.

The format of a CCL block is as follows:



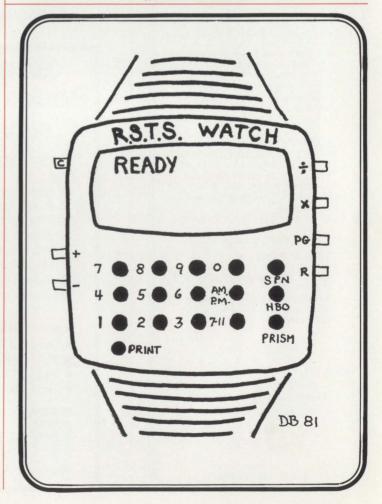
Offset O	Symbol	Description This word contains the address of the next CCL block in the linked list. This word contains a 0 for the last CCL block.
2		This word is a pointer to the first character in the CCL command text string.
4		This word is a pointer to the first op- tional character in the CCL command text string.
6		This word contains the PPN of the pro- gram to run in response to this CCL command.
8		These two words contain the file name (in RAD50) of the program to run in response to this CCL command.
12		This word contains the file extension (in RAD50) of the program to run in

IT'S 2:28 AM

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

LOTS OF LUCK! LOTS OF LUCK! SPD on Page 49

CIRCLE 80 ON READER CARD



THE RSTS EXTENSION ADDING TERMINAL LINKS TO YOUR RSTS SYSTEM

By Kevin Paul Herbert Copyright © 1981 by

Software Techniques, Inc., 5242 Katella Avenue, Suite #101, Los Alamitos, Calif. 90720

This is the first in a series of articles on adding new features to RSTS/E. Many RSTS/E users "wish" for enhancements to the operating system, either at DECUS or by sending DEC SPR forms, and often, these changes are incorporated into future versions of RSTS. More often, however, these wishes, despite their usefulness, are considered to be not worth the effort to develop and support, and are therefore not implemented. In this series of articles, I plan on providing patches to RSTS/E V7.0 that will make your RSTS system much more useful. Note that use of any of these patches will void your software support warranty. In addition, note that although every attempt has been made to insure that these patches do indeed function, no guarantee is made that these patches actually do work. With this in mind, welcome to The RSTS Extension.

Many system managers desire the capability to spy on other users (monitor all their terminal activity without being detected) or be able to communicate with other terminal users without the need to run a TALK program (so that the terminal can be used for both the communication of user messages and the demonstration of program runs and development.) In addition, many users would like additional functionality from keyboard "control" keys.

In this article, I will describe the addition of several terminal service enhancements. Five new "control" keys have been added to the terminal service. First, Control/E is a programmable function key. It is possible to program up to 32 characters to be entered to the system with the depression of the Control/E key. This function can be used to enter common responses (such as YES or NO), or to execute a common system command (such as SYSTAT). Continuing, Control/F can be used to list all of your job's open files. Although the Control/F code is listed in a previous version of the RSTS Professional, the original version conflicts with changes made to support the new features described in this article. To optionally add Control/F support and the changes detailed in this article requires a copy of TTOPNF.MAC written by Steve Davis. TTDVR.TEC is not needed, and will fail (creating an unusable monitor) if an attempt is made to use it with these changes. Further, Control/W can be used to erase the last word typed. This key is a compromise between Rubout, which erases the last character, and Control/U, which erases the last line. Control/X will cancel all un-processed type-ahead. This key can be used as a "panic" key to abort any unwanted type-ahead but not abort the current running program (as a Control/C would do). Finally, Control/Y allows a privileged user to detach his current running job.

The format for the call to program the Control/E key is as follows:

Where STRING.TO.PROGRAM\$ is a 1 to 32 character string. All control characters except for < LF> and < CR> are ignored. Note that because of an idiosyncracy in Basic-Plus, the trailing semi-colon is required. Also note that it is valid to program Control/E for other terminals that you own by including a channel number in the write.

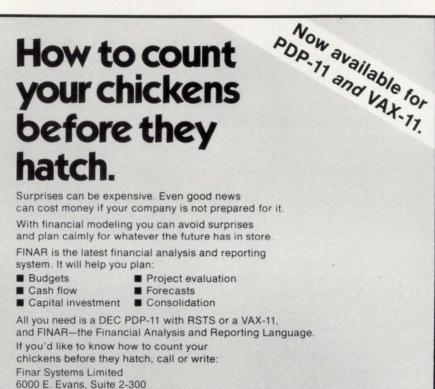
In addition to the new control keys, terminal links have been added to the monitor. With terminal links, it is possible to spy and talk to other users without the overhead of a communication program. In addition, both users are free to work normally while a spy or talk link is initiated. When talking, both users see each others output, and when spying, the spying user sees the spyed-upon user's output, but the spyed-upon user is unaware of this operation. The capability to TALK is non-privileged (if the receiving user allows users to TALK to him) and the SPY capability is privileged.

I have supplied a utility to control the terminal link capability (TLU). It is also possible for user programs to use the terminal link calls in their own programs. The following text describes the basic mechanics of "terminal links", and the use of the new utility calls.

To add these new features to the terminal driver required adding additional data storage to the monitor. This was accomplished through modifications to TTDINT.MAC. It was necessary to add storage both on a system-wide basis and a per-terminal basis. Each terminal has a device data block (DDB). The DDB contains all of the information needed by the monitor and terminal driver to control a terminal interface on a per-terminal basis. In order to enhance the terminal service, the terminal DDB has been expanded by three words (I confess, this does reduce the maximum number of small buffers possible at a rate of one small buffer per 5.33 terminals). The first of these three new words is called TTFUNK. This word contains either a zero, indicating that Control/E is not in use, or points to a small buffer containing the function key text associated with Control/E. The second word is called TTLINK. This either contains a zero, indicating that the terminal is not linked, or points to a small buffer containing internally coded pointers and flags, and 26 keyboard numbers (it is possible to link to up to 26 users). This small buffer (the link table) is shared by all members of a link group ("conference" links use no more small buffer resources than do "one-on-one" links.) The third word added to the monitor is TTLNKS. This word contains the "talk status" (defined in the next paragraph), internal flags, and a user scratch pad byte (defined later). The modification to add storage for the terminal driver on a system-wide basis consisted of adding one word (LNKLST) to the monitor. This word contains the root of the linked list of all the link tables currently in use. Although the primary purpose of LNKLST was for debugging during development, it would be possible, by studying the link table layout in LINKS.MAC, to write a program to give a link status report. LNKLST is always located at the address TTYHCT + 2 in the monitor. The value for TTYHCT is returned by the UU.TB2 (Get Monitor Tables Part II) UUO (SYS call). If there is interest in this application, I will consider writing such a utility in a

future article.

Every terminal has an associated "talk status". This "talk status" determines how the monitor handles the create a link call. The currently defined talk states are OFF, QUERY, and ON. If a user's talk status is OFF. the monitor will return an error to any user attempting to create a link (unless privileged). The user with the talk status of OFF receives no notification of the attempt, and is completely unaffected. If a user is in the OUERY state, the monitor will return an error (identical to the OFF state), although the user in QUERY state will be notified that an attempt to link was made. If a user is in the ON state. the monitor will allow any user to create a link, and will display a notification message upon the creation of the link (unless inhibited by a privileged user).

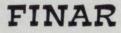


CIRCLE 51 ON READER CARD

RSTSPROFESSIONALRSTSPROFESSIONA

Denver, CO 80222 • (303) 758-7561

New York • (212) 222-2784 Chicago • (312) 876-1081 Houston • (713) 960-0848 San Francisco • (415) 956-1178 Toronto • (416) 245-8473



users can select four flags in the link creation. They are: bit O to inhibit the monitor from announcing the creation of the link, bit 1 to ignore the destination user's link status, bit 2 to spy on a user, and bit 3 to not notify (query) a user if the destination's talk status is set to query. Any of these bits can be used in combination, although there are illogical combinations (for example, spy implies ignoring talk status and

no notification, and ignoring talk status means that a user will never be queried).

Function code 4097% is the break all links call. This call takes no arguments or flags, and removes a user from any links he might have, Function code 4098% is used to set talk status. The argument is the talk status to set and is a bit oriented field. If bit O is set, a user's talk status is "on" and any user may link to this user. If bit 1 is set, a user's talk status is "query", and this user can not be linked to, but will be notified when any other user makes an attempt to create a link. If bit 2 is set (the bit is privileged), this terminal's talk status will "stay" constant across jobs and can only be changed by a privileged user. If the "stay" bit is not set, a terminal will be set to the system default talk status when the

The general format of a terminal link call is:

RESULT% = SPEC%(FUNCTION%, ARGUMENT% + SWAP%(FLAGS%), 0%, 2%)

There are four functions. They are link to a user, break all links, set talk status, and return talk status. Function code 4096% is the create a link call. The argument is the keyboard number to create a link to. In addition, privileged job is created. Bits 3 through 5 are reserved for future use, and bits 6 and 7 are reserved for internal housekeeping flags (they can never be set and will always be read as zero). Any bits can be set in combination, but the ON bit overrides the QUERY bit. Note that if neither ON nor QUERY are set, the user's talk status is OFF, and any attempts to link to the user will be denied, giving the caller an error message (unless overridden by a privileged user). The flags consist of an eight-bit number that is never used by the monitor and can be used to communicate a terminal type value to application programs. Finally, function code 4099% is used to return talk status. There are no arguments or parameters, and the result is the talk status as set by the set talk status call. For further information on these calls, see the TLU program.

Adding these new features to your RSTS monitor is a fairly simple procedure. The most difficult step will be doing the new SYSGEN to build a monitor with the enhanced version of the terminal service. For detailed information on system generation, see The System Generation Manual. The complete procedure for adding terminal links is detailed below.

Step 1 — Copy all necessary files from your RSTS/E distribution.

If your RSTS/E SYSGEN distribution is on tape, mount the tape write-locked on a magtape unit. If your distribution is on disk, mount the disk write-locked. Log in to your privileged account (the account where you would like to keep the patched version of the terminal service). For disk distributions, it will be necessary to logically mount the pack (with the mount command). The pack ID will be either SYSGNG (for RSTS/E V7.0-07) or SYSGNH (for RSTS/E V7.0-08). Run PIP and execute the following commands (note that this example assumes that your distribution is located on MTO:, if this is not true, change all references to MTO: to the appropriate device).

RUN [1.2]PIP *SY:= MT0:[1.2]TTDVR.MAC *SY:= MT0:[1.2]TTDINT.MAC *SY:= MT0:[1.2]TBL.MAC *SY:= MT0:[1.2]COMMON.MAC *SY:= MT0:[1.2]KBDEF.MAC *SY:= MT0:[1.2]KBDEF.MAC *SY:= MT0:[1.2]CHECK.MAC *SY:= MT0:[1.2]CHECK.MAC *SY:= MT0:[1.2]CHECK.MAC *SY:[1.2] = MT0:[1.2]ERR.STB *SY:[1.2]/PR:NOWARN = MT0:[1.2]SYSGEN.SAV *SY:[1.2]/PR:NOWARN = MT0:[1.2]SYSGEN.SAV *SY:[1.2]/PR:NOWARN = MT0:[1.2]SYSBAT.SAV *IZ (Control/Z to Exit)

Step 2 — Run the SYSGEN program

SYSGEN will ask questions dealing with your system's hardware configuration. For information on SYSGEN, see the System Generation Manual. Do not run SYSBAT when finished with SYSGEN, you DO have special requirements.

RUN [1,2]SYSGEN .. Answer all questions to create a monitor ...

Step 3 — Install the patches

RUN [1,2]CPATCH CPATCH V7.0-07 RSTS V7.0-07 Softec Dev 11/70 File to patch - TTDVR.MAC = TTDVR.MAC #TTDVR/CS:20188 ... Patch will be installed ... #17 (Control/Z to return to previous question) File to patch -TTDINT.MAC = TTDINT.MAC #TTDINT/CS:18991 ... Patch will be installed ... #1Z (Control/Z to return to previous question) File to patch - SYSGEN.CTL = SYSGEN.CTL SYSGEN/CS:20746 #1Z (Control/Z to return to previous question) File to patch - 1Z (Control/Z to exit)

The next step is necessary only if you want Control/F support in your monitor. The code necessary for Control/F is not included in this article, although a command file for building it in is. Note that the Control/F patch file (TTDVR.TEC) will not work properly with these changes, and will cause unpredictable results. TTDVR.TEC is not necessary for Control/F installation, although TTOPNF.MAC is. It is assumed that a copy of TTOPNF.MAC can be found in the current account. If this is not true, copy TTOPNF.MAC to the current account. It will be needed during the SYSBAT procedure. DO NOT FOLLOW THE INSTALLATION INSTRUC-TIONS THAT WERE SUPPLIED WITH CONTROL/F! ALL THAT IS NECESSARY FOR THIS CONTROL/F INSTALLATION IS THAT A COPY OF TTOPNF.MAC BE LOCATED ON THE CUR-RENT ACCOUNT. THE REST IS AUTOMATIC!

Step 4 (Optional) — Add Control/F support

RUN [1,2]CPATCH CPATCH V7.0-07 RSTS V7.0-07 Softec Dev 11/70 File to patch - SYSGEN.CTL = SYSGEN.CTL #SYSGCF/CS:36018 #12 (Control/Z to return to previous question) File to patch - 1Z (Control/Z to exit)

The next step is to run the SYSBAT program to create a monitor. Follow the procedure in the System Generation Manual for use of the SYSBAT program. After running SYSBAT, shut down your system and install the new monitor. The new system should run normally, and many new features should be available. At this point, you may install any of the new feature patches using ONLPAT (the one for Control/F and non-privileged users WILL work) and build the TLU program (if desired). Note that the TLU program should be stored with a protection code of < 104 > . The procedure for building TLU is as follows:

RUN [1,2]CSPCOM CSP>TLU.OBJ/OBJ = TLU.B2S CSP>1Z (Control/Z to exit) RUN [1.2TTKB TKB>@TLU RUN [1,2]PIP *[1,2]<104> = TLU.TSK ·1Z (Control/Z to exit) 12-Nov-81 16:37 [1,13] TL0303.001 I TL0303.001 - Make Control/Y function for non-privileged users File to patch? Module name? TER Base audress? ...PRCY Offset address? 0 Base Offset Old New? ????? 000000 001405 ? 240 ?????? 000002 ?????? ? °C 12-Nov-81 16:37 [1,13] TL0303.002 ! TL0303.002 - Set system wide default talk status ! Replace the "n" below with the desired default talk status. ! Note that this patch will not effect terminals with the ! "stay" characteristic set. File to patch? Module name? TER Base address? ...

 Base address?
 DFTK

 Offset address?
 0

 Base of cliset
 01

 New?
 2?????

 000000
 000001

 ??????
 000002

 ??????
 °C

12-Nov-81 16:38

[1,13] TL0303.003

Page 1

Page 1

Page 1

1 Tu0303.003 - Set privileged talk states. 1 Replace the "n" with the sum of the privileged talk states. 1 If it is desired to make this call privileged, replace the 1 "n" with a -1. File to patch? Module name? TER Base aodress? .KSPR Ofiset address? 0 Base Offset 01d New? 72???? 000000 000004 ? n. ?????? 000000 000004 ? n.

December 1981

page 96 RSTSPROFESSIONALRSTSPROFESSIONA [1,13] TL0303.004 12-Nov-81 16:37 Page 1 20001 \$ Start of MATN & ! TL0303.004 - Change a terminal's default talk status ! Note: For this to work, the STAY bit must be set. This implies ! that non-privileged users will be unable to change the talk status ! Change the "n" below to a keyboard number, and the "m" below ! to the desired talk status. . 2 GOTO 32767 IF E0% 6 PRINT "Tlu>"; 5 GET #0% 6 MOVE FROM #0%, CMDS=RECOUNT 6 CMDS=EDITS(CMDS,-1%) 6 I If this was a CCL entry, exit now. 5 I Prompt for input. 6 I Read a line from the terminal 6 I Get it into CMDS. 6 I Trim off any garbage. 6 2010 I to the desired talk status. File to patch? Module name? RSTS Base address? KBDDDB+(n.*DDS.KB) Ofiset address? TTLNKS Base Offset Old New? ????? ?????? 00001 ? m.!4 ?????? ?????? ????? ? C PRINT IF CCPOS(0%) & GOTO 2010 UNLESS LEN(CMD\$) & RESTORE & 2020 12-Nov-81 16:38 [1.13] TLU.B2S Page 1 Gord 2010 UNLESS LERIGHDY, RESTORE & UNTIL 08 6 READ CMD.CHK\$,SIG%,PLAGS%,VECTOR% 6 IF (SIG% AND NOT.PRIV%)>=0% 6 THEN SIG%=-SIG% IF SIG%C0% 6 MATCH%=NMATCH%(DNS,CMD.CHK\$,SIG%) 6 IF MATCH%=NMATCH%(DNS,CMD.CHK\$,SIG%) 6 IF MATCH% 6 ON VECTOR% GOTO 3000,3100,3200,2020,3300,3400,3500,3600 6 Return the carriage if needed. 6 I Take input again if a null string. 6 I Return the carriage if needed. 6 I Take input again if a null string. 6 I Return the carriage if needed. 6 I Take input again if a null string. 6 I Return the carriage if needed. 6 I take input again if a null string. 6 I Return the carriage if needed. 6 I take input again if a null string. 6 I Return the carriage if needed. 6 I Get the basig, chars if necessary. 6 I Check for a match. 6 I Save only the argument to the command. 6 I Go to the handler for that command. 6 TLUS Title: 11 Version: V7.0-02 & Edit date: 20-Oct-81 & Written by. Kevin Paul Herbert & Date: 14-0ct-81 F Package: Terminal Links & Description: Terminal Link Utility Program & Copyright (C) 1981 & Software Techniques & Los Alamitos, CA 90720 & 111 5 This software is provided free of charge to readers of the RSTS & Professional and may be copied only with the inclusion of the & above copyright notice. This software, or any other copies & thereof, may not be provided or otherwise made available to any & other person except for non-commercial use and to one who agrees & to these license terms. Tile to and ownership of the software & shall at all times remain in Software Techniques. & 2030 1 Finish the loop. & PRINT "Unknown command - "+CMD\$ & PRINT "Type 'HELP' if you need it." UNLESS E0% & GOTO 2010 & 2040 The information in this document is subject to change without & notice and should not be construed as a commitment by Software & Techniques. & 07CO 2010 & Report an error. & Tell the user about HELP unless CCL entry. & Go back for more. & 1 Meport an error. *
1 Tell the user about HELP unless CCL entry. 4
1 Go back for more. 4
1 This handles the TALK, SPY, and LINK commands. 4
1 The tark command is used to create a link. 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command (privileged) is used to TALK ignoring 4
1 The tark command ignored to the tark ignoring 4
1 The tark command ignored to the tark ignoring 4
1 The tark command ignored to the tark ignoring 4
1 The tark command ignored to the tark ignored to tark ignored to tark ignored to the tark ignored to the tark ignored to tark ignor This software is un-released and Software Techniques has no & commitment to support it at this time, unless stated elsewhere in & writing. & 3000 201 8 Modification History & Ver/Edit Date Reason (Who) & Initial conception. & Fixed STAY problem. (KPH) & Updated for LINKS V2.1-04 & V7.0-01 V7.0-02 14-Oct-81 20-Oct-81 21 ! 1001 8 Program Description & This program allows a user to create and break terminal links. In addition, it allows the privileged user to spy on other users. 5 4001 Variables and Arrays 5 5001 5 Compile time variables & 1 2 Derine .VersionS = "V7.0-02" & Define .Create.Link% = 4096% & Define .Ignore.Status% = 512% & Define .Break.Link% = 4097% & Define .Break.Link% = 4097% & Define .Get.Link% = 4098% & Define .Get.Link% = 4098% & Define .Talk.Oury% = 2% & Define .Talk.Query% = 2% & Define .Talk.Stay% = 4% & I Program Version. & I Create a link call & 501

3100

3200

3210

1

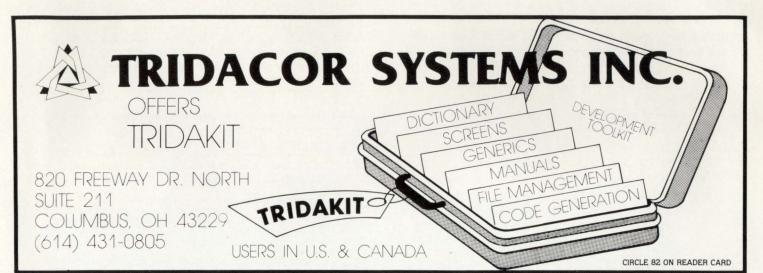
7501 6 Functions & Line # Name Description & FNMATCH& 14000 Check for string matches. & 9001 Dimension Declaration & 6 901-929 local dimension declarations 6 930-949 library dimension declarations 6 950-979 MAP statements 6 MAP (KEYWRD) KEYWRD% & I Define the keyword. & I Por this to work, the following line must be added & I into the TKB command file: & I VSECT=KEYWRD:400:2 & 950 9991 8 Start of Initialization & 5 Onerror Goto 19000 & ! Set standard error trap. & 1000 Print "Tlu " + .Version\$ + " Software Techniques" & + CR + LF + "Terminal Link Utility"+ CR + LF & 1010 Unless E0% & ! Print standard header on 'Run' entry. & ! Derine various variables. & NOT.PRIV%=(KEYWRD% AND 1024%)=0% & I\$=SYS(CHRS(6%)+CHRS(-21%)) & GOTO 2020 IF E0% & ! Set up the non-priv flag (-1 for non-privileged users.) & ! Get rid of privileges (we shouldn't have them anyway.) & ! Skip the first-time prompt if CCL entry. & 1030

I Go back for more. 6
I This is the STATUS command. 6
I Th will list the user's current talk status and display 6
I the terminal type byte if it is non-zero. 6
STS%=SPECE(, GET.LINK*,0%,0%,0%,2%) %
PHINT "Talk status is "; "
IF STS% AND .TALK.ON% 6
THEN PRINT "On." %
ELSE IF STS% AND .TALK.OUERY% 6
THEN PRINT "Outry." 6
I Get the talk status is... 6
I If ont fuery. 6
I If ont query. 6
I If not query. 6
I Say "Oft." 6
PRINT "Talk status stays between jobs." IF STS% AND .TALK.ST

! This is the HELP command. &
! We will print out how to use this program. &
PRINT "Commands are:" + CR + LF + &
 HT + "TALK n" + HT + "Talk to keyboard N" + CR + LF + &
 HT + "BREAK" + HT + "Break all links" + CR + LF + & 3 400

! This is the BREAK command, &
! It is used to break all of a user's links (or SPYs). &
Is=SPEC\$(.8REAK.LINK\$,0%,0%,0%,2%) &
GOTO 2010 &
! Break all links. &
! Go back for more. &

RSTSPROFESSIONALRSTSPROFESSIONA



+ HT + "Get talk status" + CR + LF + 6 + HT + "Set talk status to ON" + CR + LF + 6 + HT + "Set talk status to OPE" + CR + LF + 6 + HT + "Set talk status to QUERY" + CR + LF + 6 + HT + "Talk regardless to keyboard N" + CR + LF + 6 + HT + "Talk regardless to keyboard N" + CR + LF + 6 + HT + "Spy on keyboard N" + CR + LF + 6 + HT + "Set STAY characteristic" + CR + LF + 6 + HT + "Reset STAY characteristic" 6 S NOT_PRIV& 6 "STATUS" "ON" "OFF" "QUERY-" "TYPE" "LINK n" "SPY n" "STAY" HT + PRINT HT HT + "UNSTAY" UNLESS NOT. PRIV& & GOTO 2010 & Define the non-priv'd commands. & Derine the priv'd commands unless non-priv'd user. & Go back for more. & 3400 1 1 This is the 'STAY and 'UNSTAY' command. \$
1%=SPBC%(.Set.Link%,(SPEC%(.Get.Link%,0%,0%,2%) AND -5%) OR FLAGS%, \$
0%2%) \$
GOTO 2010 \$
1 Turn off the stay bit. \$
1 Go back for more. \$ 3500 3600 I Get the type value. & I If it is out of range, & I Give an error. & I Go back & I Go back & I Go back & I If it seems in range, & I If it is unspecified, & I Prompt for it. & I Take input. & I Move it into the buffer & I Trim off any junk. & I from othing was specified, & I forbing was specified, & I forbing was specified, & I Go back & I Set the new value & I Go back & I Go back & Goto 32700 & ! Prevent fall through errors. & 9999 14000 1 & Local Functions & DEF FNMATCH%(COMMANDS,CHECKS,MATCH%) 5 FOR 20%=LEN(CHECKS) TO MATCH% STEP -1% 6 IF LEFT(COMMANDS,20%)=LEFT(CHECKS,20%) 6 THEN FNMATCH%=20%+1% 6 NEXT 20% & FNMATCH%=0% & PNEND & ! Continue the loop. & ! Set failure. & ! Exit the function. & 14010

19000!	8	
1	Error Handler &	
1.0000		
19003	IF ERR=3% & THEN IF ERL=3000% & THEN PRINT "?Link entry already in use" & RESUME 2010 &	
	! If "INUSE" 6	
	<pre>! If at the create a link, &</pre>	
19004	IF ERR=4% &	
	IF EXE-46 % THEN IF ERE-3000% & THEN PRINT "?No room in link table" & N RESUME 2010 & I If "NORCOM" &	
	! If "NOROOM" &	
	<pre>! At the create a link, & ! At the create a link, & ! Report this (will probably never h ! Go back for more. &</pre>	appen). &
19006	Ir ERR=6% & THEN IF ERL=3000% &	
	THEN PRINT "?Keyboard number out of range RESUME 2010 &	* 6
	! If "NODEVC" & ! At the create a link, &	
	! Report KB number out of range. & ! Take more input. &	
19007	IF ERR=7% &	
	THEN IF ERL=3000% & THEN PRINT "?Link already established" & RESUME 2010 &	
	! If "NOTCLS" & ! If at the create a link, &	
	! If at the create a link, & ! Report the error. & ! Go back for more. &	
19008	IF ERR=8% &	
19000	THEN IF ERL=3000% & THEN PRINT "?There is no job running on t RESUME 2010 &	hat terminal" &
	I IF "NOTAVI" &	
	I At the create a link, & I At the create a link, & I Tell the user why we couldn't link I Take more input. &	
19010	IF ERR=10% & THEN IF ERL=3000% &	
	THEN IF ERL=3000% & THEN PRINT "?Terminal not receiving links RESUME 2010 &	• 6
	ELSE IF ERL=3400% or ERL=3600% & THEN PRINT "7Talk status is set to STAY" RESUME 2010 &	
	RESUME 2010 &	
	! If PRVIOL & ! At the create a link, &	
	! At the create a link, & ! Tell the user that talk is off. & ! Take more input &	
	<pre>! At the set link status or terminal type by ! Tell the user he can't change talk ! Take more input. &</pre>	te, & status. &
19011	If (Err = 11%) & Then Resume 32700 &	
	Then Resume 32700 & ! End of file on device. &	
19018	IF ERR=18% &	
	THEN IF ERL=3000% & THEN PRINT "?You can't link to yourself" RESUME 2010 &	
	I If "BADFIIO" &	
	I At the create a link, & I Tell the user of his silliness. & I Go back for more. &	
19032	IF ERR=32% & THEN IF ERL=3000% &	
	THEN IF ERL=3000% & THEN PRINT "?No room to create link table RESUME 2010 &	* 6
	! If the V7.0 nemesis, &	
	! At the create a link call, & I Report it nicely, & I Go back for more &	
19052	IF ERR=52% & THEN IF ERL=3000% & THEN PRINT "?Illegal keyboard number" &	
	and a stand a standar heleoard humber a	

page 98 RSTSPROFESSIONALRSTSPROFESSIONA <cc> <esc='0AV<cr> 605:(tab)CALL(tab)TTSJST<tab>(tab);;G0 SFT THE OUTPUT READY FLAG<cr> H/TTWIPE::/V<cr> TWIPE::CALLX*tab>CLREUF,R5,DDBUPC+EP ;;NOW WIPE OUT OUTPUT BUPFER<cr> RESUME 2010 6 ELSE IF ERL=3600% 6 THEN PRINT "?Illegal type number" 6 PRESUME 2010 6 ! If illegal number, 6 ! If processing the keyboard number, 6 ! Give an error. 6 ! Give an other command. 6 ! If processing the type byte, 6 ! Give an error. 6 ! Get another command. 6 RESUME 2010 & *I<cr>
CALL<tab>@EXELNK<tab><tab>;;DO THE HOOK FOR LINKS (EXELNK)<cr> <tab><esc>*V<cr>
<tab>CALLX<tab>CLRBUF,P5,DDBUFC+EF ;;NOW WIPE OUT OUTPUT BUFFER<cr></tab RUBOUT/V<cr>>BIT<tab>#RUBOUT,DDFLAG(F1) ;;IN RUBOUT MODE ?<cr> cdab>BIT<tab>#RUBOUT,DDFLAG(F1) ;;IN RUBOUT ADA *0A2RV<cc> *0A2RV<cc> *Cab>CMP<tab>R2,#177<tab><tab>;;WE WERE, SHOULD WE REMAIN<cc> *G/;;/KI<cc> IS THIS A RUBOUT?<cc> <esc>*V<cc> <tab>RUBOUT?<cc> <tab</tab> IF ERR=57% & THEN IF ERL=2020% & THEN RESUME 2040 & THEN RESUME 2040 & ! If we got out of data, & ! When we read the commands. & ! When we read the commands. & ! Go back to report the error. & 1905/ 19999 ctabSCALL<tab>@TTIDSP=~TTICHL+2>(R3) _;;EF, GO PUSH SPECIAL ROUTINE<cr>
<tab>Rticer>
<tabRticer>
<tabRticer</tabRticer>
<tabRticer>
<tabRtiter</tabRtic 200001 6 Data Statements & "ON", "OFF", "QUERY", "STAY", "TYPE", "TALK", "SPY", "LINK", "BREAK", "STATUS", "HELP". DATA 2, 2, 1, 4, 2, 3, 1, 6, 6, 7, 8, 1, 1, 0, 2, 4, 0, 0, cess=V<cr> ctabsRctabsTTIN04<tab>;;NORMAL THING TO DO NOW IS NOTHING...<cr> +H/TTILFC:/G/305:/04<cr> 6, & 6, & 7, & 8, & 1, & 1, & Value (Value (Valu <esc>* <ff> *1<cr> -2, 1024, <cr>cubuBL<tab><LK.BSX><cr>cubuBL<tab><LK.BSX><cr>cesc>#/TTIJOB;/H/TTMODE(R1)/V<cr>cab>CLTAtabTTMODE(R1)<tab>;;NO SPECIAL NODE OR DELINITER<cr>stab>CLL<tab>THODE(R1)<tab>;;SET UP THE DEFAULT TALK STATUS<cr> 2, 8 1, 2, 1, 0, 0, 0, "HELP", 0, 8 <esc>*H/ENDRUB::/V<cr>
ENDRUB::BIC<tab>#RUBOUT,DDFLAG(P1) ;;END THE RUBOUT MODE<cr> "UNSTAY". -1' *Icer>
BIT<tab>#RUBOUT,DDPLAC(F1) ;;KERE WE IN RUBOUT WODE?<er>
ctab>BEQ<tab>10\$<tab><tab>;N0, NOTHING TO DO<er>
<tab>Ecc+ab>fRUBOUT,DDPLAC(P1) ;;END THE RUBOUT MODE<er> ! CCL ENTRY POINT & CMDS=EDITS(SYS(CHRS(7%)),-1%) & E0%=(CMDS<>*TLU*) & GOTO 1000 & ! Get core common (junking junk). & ! Set the entry flag. & ! Set the entry flag. & ! Go to the top. & 30000 M1(cr)
<tab>ELt(tab>@EXELNK(tab>(tab>;;EXECUTE THIS FOR LINKS(cr)
<tab>BIT(tab)@ITSCOP,TTCHAR(R1) ;;IS THIS A SCOPE?(cr)
<tab>BNE(tab>l05(tab>(tab>;;YES, DON'T SAY ANYTHING(cr)
<cr)</tr> (esc)*V<cr>(csc)*V<cr>(cab>Action of the state of th 32/00! & Completion Routines & 8 32710 Close 1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%, 11%, 12% & ! Close all channels (the fast way). & CtableControl and a control and a contr 32/67 Eng 12-Nov-81 16:58 [1.13] TLU. CMD Page 1 SY:TLU=SY:TLU, LB:CSPCOM/LB / UNITS = 12 ASG = SY:5:6:7:8:9:10:11:12 ;RESLIB=LB:BASICS/RO VSECT=KEYWRD:400:2 TASK=...TLU EXTTSK= 512 *#/AETURN/V<ct> *#/AETURN/V<ct> <tab>ETURN/V<ct> <tab>ETURN/V<ct> <tab>ETURN/V<ct> *AI<<tr> *Al<cr>
90\$:<tab>JMP<tab>CHOUTE<tab><tab>;;GO AND OUTPUT A CHARACTER<cr>
100\$:<tab>JMP<tab>NOCHAR<tab><tab>;;NO MORE CHARACTERS TO RUB OUT<cr> <cr> ----- TT-VE.-N 12-1.00--1 20:1 Page 1 *AV-cr-TITL.ctai-TTUVE, TERMIDAL DEIVEE ,0",22-SEP-75,MBB,JDN/WEN/GW/APC SJK/ETV/AWP-cr *G/22-SEP-79/-9G/CR-CCT-R1/G/ALP/IT,EPI-GAV-cr TIL.ctai-TTUVE, TERMITAL DEIVEE ,07,08-CCT-R1,MHB,JDN,WEN/GW/APC,SJK/PTW/AWP/KPH/SPD-cr *H/TTMATP/DAV-cr *tab:DEPORG-tab-TTWAPP-cr *iccr cr: tab>DEFORG tab TTDVEA tab ;FNTRY POINT VECTORS FOR THEFINAL LINKS or ctab.okulctab.WEXELNK<tab><tab>;tAKE SURE WE HAVE NO LIN 50\$:<cesc.*V<cr> 50\$:<tab>DECB<tab>DDHOR2(R1)<tab>;;YES, WILL IT PIT?<cr> *G/40\$/-3C/50\$/V<cr> <tab>Rtcab=St6xtab>50\$<tab>tab>tab>tab>tr AGAIN<cr> *H/CHROUT::/<cr> tab DEFORG tab TTOWRD tab ; TEPMINAL LINK CODE WILL GO HEFE or (tab)DEFORG tab)TTDVRR tab ; RUBOUT CODE WILL CO HERE (AND BE OVEPLAID) or esc>*2AI (cr) cr> tat=ORG<tab>TTMAFX<tab -: MAPPED CODE EXTENSION (FOR LINKS)<cr TICCT) CALL(tab)@EXELNK(tab)(tab);;MAKE SURE THIS WORKS FOR LINKED TERMINALS(cr) (tab)@ExeS'H/CHKFRE -/V(cr) ; CHKFRE - CHECK FOR OUTPUT BUFFER AVAILABILITY.(cr) * cr <esc >* 0AV < cr <tab>ORG<tab >TTSYST < cr *AI < cr</pre> *Aicc: cr: ctr: RECHKF - RE-CHECK FOR BUFFERS (IGNORING LINK FLAGS) <cr> esc>*V(cr) ;<cr>*2Alccr>
*2Alccr>
;<tab>CALL<tab>RECHKF<cr>
;<cr>
;<cr>
<esc>*V<cr> *AllK1<cr> .ENABL<tab>LSB<cr> .ENABL(tab)LSB<Cr>

CF2
CHXFFF:(ALL(tab)@CHKLNK,F5,IK,BUF ;;LINK OUT CF BUFFERS?
CF3
CHXFFF:(ALL(tab)@CHLNK,F5,IK,BUF ;;LINK OUT CF BUFFERS?
CF4
Ctab>EDECtab>205(tab)
Ctab>205(tab)
Ctab>2 *1<cr>

(tab)CALL+tab>@CLRLNK,P5,LK.PSX ;;CLEAR THE LINEFEED NEEDED BIT<cr`</td>

(esc)*V<cr>
(esc)*V<cr>
(tab)SPLC+tab>3<tab>;;NOW WE CAN GO BACK TO LEVEL 3<cr>

*G/GLOBAL+tab>/G/IDEXIT/I/,LK.PSX/V<cr>

GLOBAL+tab>/G/IDEXIT/I/,LK.PSX/v<cr>

</tab> GLUBAL<tab><JBWAIT,IOEXIT,LK.PSX><cr>
*#/DETJOB:/AV<cr>
ctab>CLRB<tab>DJBNO(P1)<tab>;;NON "DETACH" THE JOB<cr>
i<cr>

</tab> CLab.CLRB.CLab.PBRKLNK.Lab.cLab.;;sew Delken in Johnet's Johnet's
clab.CALL(tab>@BRKLNK.Lab.cLab.yBREAK ANY LINKS TO THIS TERMINAL<cr>
cesc>%HORDNT:/H/905:<Lab.CALL/Vecr:
905::Lab.CALL/Lab.TBS.CLAb.YCL/Vecr:
905::Lab.CALL/Lab.TTSJSTCLab>clab.YCLS
905::Lab.CALL/Lab.TTSJSTCLab>clab.yclBeCk FOR RESTARTING JOB<cr>
%H/SLINFD::MUV:Lab.YTTJSTCLab>(tab);:CHECK FOR RESTARTING JOB<cr>
%LSLNTD::MUV:Lab.YTTJN,-(SP)(tab);:SET UP TO GO TO "TTIN"<cr>
%Loc: .DSABL<tab>LSB<cr *I<cr> MOV<tab>*LHKLIN,-(SP)<tab>;;MAKE SURE THAT CHKLIN GETS CALLED (FOR LINKS)<cr> GLOBAL<tab><LK.BUF,BPQ.FB - cr <tab.vesc.*V<cr><tab.vesc.*V<cr><tab.wesc.*V<cr><tab.wesc.*V<cr><tab.wesc.*V<cr><tab.wesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc.*V<tab.vesc <fi>

Sesc>*V<cr>

SBTTL<tab>Form FEEDS AED VEPTICAL TABE ON IMPUT(cf) SBTID:tab/rown ribbs Abs thereing the rest of the rest of MyCHREDE::/ifcr CALL:tab/GSPYING:tab/tab/:SHOULD WE DC THIS FOR LINES?/cr Ctab/SCALL(tab/SCALL):tab/:SNO/cr ctab/SCALL(tab/SCHELNE,P5,LE,PSX ;;DOES THIS LINE NEED A LP?/cr

<cr>
GLOBAL<tab><RTSPC><cr>

<cr>
<tab>.DSECT<tab><tab>;DEFINE THE NEW FUNCTION KEYS<cr>

<cr>TisC::tab>.BLKWtab>tab>:THIS IS FUNCTION CODE 0cr>TisC::tab>.BLKWtab>tab>:tab>:tab>:tab

<cr><tab>UNORG<cr>

*AI<cr *9Kicc> <tab>CALL<tab>CHKINC<tab><tab>;;CHECK TO SEE IF INPUT CHARACTERS ALLOWED.<cr> <seb>V(cr> <tab>CMP(tab>R2, **=100<tab>;;IS IT CONTEOL/R?<cr> CHEVIC - CHECK IF SPECIAL INPUT CHARACTERS ARE ALLOWED (NO ECHO CTL ALLOWED)(CT> CHEMEC - CHECK IF SPECIAL INPUT CHARACTERS ARE ALLOWED (ALLOW ECHO CTL)(CT> <tab>CALL<tab>CHKINC<cr><tab>CALL<tab>CHKINC<cr><tab>CALL<tab>CHKNEC<cr> $\langle ctc \rangle$ $\langle tab \rangle RETURN IS MADE TO CALLER IF SPECIAL FUNCTIONS ALLOWED, <math display="inline">\langle ct \rangle$ $\langle ctab \rangle$ RETURN IS TO TINORM IF SPECIAL FUNCTIONS NOT ALLOWED, $\langle cr \rangle$ <cr>> .ENABL<tab>LSB<cr> <cr>
CHKINC::<cr> <cr>
.IF<tab>NE<tab>EKOCTL<cr> <tab>BITB<tab>#TTECTL,TTMODE(R1) ;;ECHO CONTROL ACTIVE?<cr><tab>BNE<tab>l0\$<tab>;;YES, MAKE THIS A NORMAL CHARACTER<cr></tab> <cr>
.ENDC<tab>;EKOCTL<cr> (cr) CHENEK::CALL<tab>TTISGL<tab>(tab>(tab);CHECK POR SPECIAL MODE(S)<cr> CHENEK::CALL<tab>TTISGL<tab>(tab);CHECK POR SPECIAL MODE(S)<cr> (tab>BRC<tab)10\$(tab)<tab);THARE NORMAL CHARACTER IF NNY<cr> (tab>BRC(tab)20\$(tab)<tab);THES, DO THE NORMAL EXIT(scr) 10\$:(tab)285(tab)20\$(tab)<tab);THES, DO THE NORMAL EXIT(scr) 10\$:(tab)TBTCHab)<tab);THES, DO THE NORMAL EXIT(scr) 10\$:(tab)TBTCHAb)TBTOR(tab)<tab);TBND MAKE THIS A NORMAL CHARACTER(scr) 20\$:(tab)ERC(tab)>(tab);TBND MAKE THIS A NORMAL CHARACTER(scr) 20\$:(tab)ERC(tab)<tab);TBSPECIAL CHARACTERS ARE 0K.<cr)</pre> .DSABL<tab>LSB<cr> <esc>*H/CHKLIN::/V<cr>CHKLIN::TST<tab>DDPLAG(R1)<tab>;;IS LINE STALLED FOR PAGING?<cr> *ICcr> MOV(tab)4405,-(SP)<tab>;;ENSURE THAT CSR POINTER IS RETURNED<Cr> <tab>CALL(tab)EEXELNK<tab>;;DO THE LINKS HOOK<cr> <tab>Cab>BK(tab)SA(cr> <tab>BK(tab)S(tab>;;tab>;;SKIP THE NEXT PUSH (SAVES STACK SPACE)<cr> *G/SRTLIN::/V<cr> SRTLIN::MOV<tab>CSR.KB(R0),R3<tab>;;UNCONDITIONAL LINE STARTUP<cr> SRTLIN::MOV(tab)CSR.KB(R0),R3(tab);JUNCONDITIONAL LINE STARTUP(CT) *1ccr> MOV(tab)405,-(SP)(tab);JENSURE RETURNED CSR POINTER IS CORRECT(CT) (tab)CALL(tab)@RELNN(tab)(tab);JENTOTHE HOOK POR LINKS(CT) 55;(tab)V(stab)CSR.KB(R0),R3(tab);JUNCONDITIONAL LINE STARTUP(CT) *1ab)CALL(tab)@DEPTLK(tab)(tab);SET UP THE DEPAULT TALK STATUS(CT) @GSC)V(CT) (tab)SRR(tab)205(tab)(tab)(GD (LEAR INPUT AND EXIT(CT) *1/TTENT;0/805;100V(CT) 805;(tab)SEROR(tab)PRVIOL(tab)(tab);SECIAL FUNCTION ERROR(ct) *1(ct) PRVERR:<tab><tab><tab><tab>;wE NEED THIS IF LINKS ARE NOT LINKED IN<cr> <cr>(cr>(esc>*H/SPCSKB/G/:-/0AI<cr> :<tab)F CODE & 10000 <> 0, THE FUNCTION IS HANDLED BY "LNKSPC"<cr>;<tab>THE LINK SPECIAL HANDLER.<cr></tab> ide line black blac 85\$:<esc>*V(cr> 85\$:<tab/tab/tab/tab/fis FUNCTION A SIMPLE FUNCTION?<cr> *I/(BELL)/OAV<cr> <tab.WORD(tab>'G-100<tab>(tab)(CONTROL/G (BELL)<cr> *I<cr> *Icor>

<cab: WORD</pre>
Cab: WORD
Cab: *1(cr> <cab.wkRb<tab>'W-100<tab>;CONTROL/W (WORD DELETE)<cr> <tab.wkRb<tab>'X-100<tab>cab;CONTROL/X (CANCEL TYPE-AHEAD)<cr> <tab.wkRb<tab>'Y-100<tab>cab;CONTROL/Y (DETACH)<cr> <tab.wkRb<tab>'Y-100<tab>cab;CONTROL/Y (DETACH)<cr> <tab.wkRb<tab>'Y-10CG(OAV<cr> <tab.wkRb<tab>'Y-10CG(C+ab><tab);CONTROL/G<cr> ticct)
 (ctb).wkRo(tab)TISCE(tab)<tab);CONTROL/E(cr)
 (tab).wkRo(tab)TISCE(tab)<tab);CONTROL/F(cr)
 (ctb).wkRo(tab)TISCE(tab);tab);CONTROL/E(cr)
 (tab).wkRo(tab)TISCE(tab);tab);CONTROL/E(cr)
 (tab).wkRo(tab)TISCE(tab);CONTROL/E(cr)
 (tab).wkRo(tab)TISCE(tab);CONTROL/E(cr)
 (tab).wkRo(tab)TISCE(tab);CONTROL/E(cr)
 (tab).wkRo(tab)TISCE(tab);CONTROL/E(cr)
 (tab):wkRo(tab)TISCE(tab);CONTROL/E(cr)
 (tab):wkRo(tab)SUB(TISCE(tab);CONTROL/E(cr)
 (tab):WkRo(tab)SUB(TISCE(tab);CAB);CONTROL/E(cr)
 (tab):WkRo(tab)SUB(TISCE(tab);CAB);CAB);CAB);CAB);CAB);MA(TISCE(tab);CAB);CAB);CAB);CAB);MA(TISCE(tab);CAB);CAB);CAB);CAB);MA(TISCE(tab);CAB);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);MA(TISCE(tab);CAB);M I<cr>
SBTTL<tab>DEFINE VECTORS AND CONSTANTS FOR TERMINAL LINKS<cr> <tab>TMPORG<tab>TTDVRA<cr> (tab)THPURG(tab)TTDVMAGL; cr: cr: SETFUN:(tab).WORD(tab)RTSBC(tab)(tab);JUST DO A RETURN IF NO FUNCTION KEY(cr) CHLLMK:(tab).WORD(tab)RTSBC(tab)(tab);RTBTHING TD DO...(cr) BRKLMK:(tab).WORD(tab)RTSBC(tab)(tab);NOTHING TD DO...(cr) SFITMG:(tab).WORD(tab)RTSBC(tab)(tab);NOTHING TD DO...(cr) SFITMG:(tab).WORD(tab)RTSBC(tab)(tab);NOTHING TD DO...(cr) SFITMG:(tab).WORD(tab)RTSBC(tab)(tab);NDTHING TD DO...(cr) SFITMG:(tab).WORD(tab)RTSBC(tab)(tab);NDTHING TD DO...(cr) SFITMG:(tab).WORD(tab)RTSBC(tab)(tab);NDTH THE ARGUMENT AND RETURN(cr) SFITMG:(tab).WORD(tab)RTSBC(tab)(tab);SFI UP DEFAULT TALK STATUS(cr) DFRIME(tab).WORD(tab)PFVERG(tab)(tab);SFI UP DEFAULT TALK STATUS(cr) NUDOK:(tab).WORD(tab)PFVERG(tab)(tab);THE NEW CHARACTERS DON'T EXIST(cr) NUDOK:(tab).WORD(tab)TSAVE(tab)(tab);THE NEW CHARACTERS DON'T EXIST(cr)

page 99 RSTSPROFESSIONALRSTSPROFESSIONA

> <tab>TMPORG<tab>TTDVRB<cr> ;+<cr>
> ; RTSR5 - SKIP AN ARGUMENT<cr> (ctab>NOTE: THIS ROUTINE ONLY EXISTS (AND NEEDS TO EXIST) IF(cr) <ctab>TERMINAL LINKS ARE NOT LINKED IN. IF TERMINAL LINKS(cr) <ttab>EXIST, THIS CODE WILL BE OVERLAID (REEP THE SYMBOL(cr) <ttab)LOCAL1)(cr)</pre> <cr> xx1> RTSR5:<tab>CMP<tab>(R5),(R5)+<tab>;BUMP PAST THE ARGUMENT (AND SET 2)<cr> <tab>RTS<tab>R5<tab>{tab>;AND GO BACK TO THE CALLER<cr> (cr> (tab)UNORG(cr) ff <ff>
> <esc>*V<cr>
> .SBTTL<tab>SOME USEFUL ASCII STRINGS<cr>
> *EX<cr> -3x/512741/V<cr>
> -3x/512741/V<cr>
> -3x/512741/V<cr>
> -1F<tab>RE<tab>ST2741<<cr>
> +011
> -1F<tab>NE<tab>ST2741<<cr>
> +1F<tab>NE<tab>T01
> -1F<tab>NE<tab>T01
> -1F<tab>NE<tab>T01
> -1F<tab>NE<tab>T01
> -1F<tab>NE<tab>T01
> -1F<tab>NE<tab>T01
> -1F<tab>NE<tab>T01
> -1F<tab>NE<tab>T1N(SE<<cr>
> -1F<tab>NE<tab>T1N(SE<<cr>
> -1F<tab>NE<tab>T1N(SE<<cr>
> -1F<tab>NE<tab>T17
> -1F<tab>NE<tab>T17
> -1F<tab>NE<tab>T17
> -1F
> -1F<tab>NE<tab>T2741
> -1F
> -1F
> -1F<</p>
> -1F<</p>
> -1F<</p>
> -1F<</p> +DAIGCr> LFKtap>NEKtab>INC\$LKKCr> LNKLST::+UGROKtab>üKtab>GROCT OF LINKED LIST OF LINK TABLESKcr> .ENDCKtab>INC\$LKKCr> Kasc>fGAVCr> .IF<tab>NE<tab>022741<cr>
> *Ex<cr> 16-Nov-81 11:02 EXEC: [1,13] LINKS.MAC Page 1 Title LINKS, <Terminal Driver Extensions>,05,19-Oct-81, <KPH> Written by: Kevin Paul Herbert 15-Aug-80 Date: Package: RSTS V7.0 Enhancements Description: Code to enhance RSTS/E terminal service Copyright (C) 1981 Software Techniques Los Alamitos, CA 90720 This software is provided free of charge to readers or the RSTS Professional and may be copied only with the inclusion or the abuve copyright notice. This software, or any other copies thereof, may not be provided or otherwise made available to any other person except for non-commercial use and to one who agrees to these license terms. Title to and ownership of the software shall at all times remain in Software Techniques. The information in this document is subject to change without notice and should not be construed as a commitment by Soitware Techniques. This software is un-released and Software Techniques has no commitment to support it at this time, unless stated elsewhere in writing. Modification History .Sbttl Ver/Edit Date Reason (Who) Copy from V2.0. (KPH) Facilitate PSECT split. (KPH) Clean up code, fix many bugs. (KPH) Add conditionals. (KPH) Move RUBOUT code. (KPH) Fix STAY problem. (KPH) Final code clean-up pass. (KPH) V2.1-01 27-Sep-81 V2.1-02 01-Oct-81 V2.1-03 08-Oct-81 V2.1-04 V2.1-05 15-Oct-81 19-Oct-81 SBTTL SET UP OUR DEFAULT PARAMETERS INC\$CE, INC\$CE = INC\$CF, INC\$CF = INC\$CW, INC\$CW = INC\$CX, INC\$CX = INC\$CY, INC\$CY = INC\$LK, INC\$LK = .IIF .IIF .IIF .IIF .IIF .IIF NDF NDF NDF NDF NDF ; DEPAULT TO INCLUDE CTRL/E ; DEPAULT TO INCLUDE CTRL/P ; DEPAULT TO INCLUDE CTRL/W ; DEPAULT TO INCLUDE CTRL/X ; DEPAULT TO INCLUDE CTRL/Y ; DEPAULT TO INCLUDE LINKS INC\$CH = INC\$CEIINC\$CFIINC\$CWIINC\$CXIINC\$CY ; INCLUDE NEW CHARACTERS .IIF EQ INC\$CHIINC\$LK, .ERROR ;NO "LINK" FUNCTIONS SELECTED .SBTTL DEFINE OUR MACROS .IF NE INC\$CH DEFINE A VECTOR ENTRY MACRO DEFVEC CHAR NE INC\$C'CHAR .IF .WORD TISC'CHAR ; CONTROL/ 'CHAR .IFF

;INC\$C'CHAR

.WORD INCOMP

; DUMP THE CONTROL/ 'CHAR

page 100

December 1981

RSTSPROFESSIONAL

. ENDC	;INC\$C'C	HAR		
.ENDM	DEFVEC			
.ENDC .SBTTL	;INC\$CH DEFINE 1	ERMINAL	LINK BIT	s
.BSECT			; IN TTLN	KS [LINK STATUS]
LS.RCV:	.BLKB		; TERMINA	L LINKS ARE ENABLED
LS.QRY: LS.STY:	.BLKB	:	;QUERY U ;TALK ST	SER ON LINK FAILURE ATUS SHOULD "STAY" ACROSS JOBS
	.BLKB .BLKB		; RESERVE ; RESERVE	D
	.BLKB	•	: RESERVE	D CURRENTLY PROCESSING LINKS
LS.LNK: LS.SPY:	.BLKB .BLKB	:	; THIS TE	RMINAL IS SPYING
	.BLKB .BLKB		: RESERVE	D FOR CUSTOMER EXTENSIONS D FOR CUSTOMER EXTENSIONS
	.BLKB .BLKB	:	: RESERVE	D FOR CUSTOMER EXTENSIONS D FOR CUSTOMER EXTENSIONS
	.BLKB		; RESERVE	D FOR CUSTOMER EXTENSIONS D FOR CUSTOMER EXTENSIONS
	.BLKB	:	; RESERVE	D FOR CUSTOMER EXTENSIONS D FOR CUSTOMER EXTENSIONS
	.BLKB	•		
.BSECT	HIGH		;IN LT.F	LG [LINK FLAGS]
	.BLKB .BLKB	:	; RESERVE ; RESERVE	
	.BLKB		; RESERVE ; RESERVE	D
	.BLKB		; RESERVE	D
LK.BUF:		:	;RESERVE ;LINK IS	ESTALLED FOR BUFFERS
LK.RSX:	:.BLKB	•		
.DSECT				L LINK TABLE LAYOUT
LT.CNT: LT.FLG:	.BLKB		; COUNT C ; LINK FL	F USERS IN TABLE
LT.KB:	.BLKB	26.	:KEYBOAR	D NUMBERS IN THE TABLE NEXT @ LT.NXT
LT.NXT: LT.PRv:	. BLKW		;LINK TO	PREVIOUS @ LT.NXT
.SBTTL	ORG	TTDVRA	FIONS	;JUMP VECTOR .PSECT
	ORG ORG	TTDVRB		RESIDENT CODE .PSECT RUBOUT CODE (OVERLAYS TTDVR)
	DEFURG	TTMAPX TTOPNF		;PAR6 CODE ;OPEN FILES CODE
.SBTTL			P VECTORS	3
	ORG	TTDVRA		;SET UP THE JUMP VECTORS (FOR REAL)
.IF	NE	INC\$CE		
	.WURD	SETFUN		;SET A FUNCTION KEY
.IFF	;INC\$CE			
	.WORD	RTSPC		;NO FUNCTION KEY SUPPORT
. ENDC	;INC\$CE			
.IF	NE	INC\$LK		
	.WORD	CHKLNK		;CHECK LINK FLAGS
	.WURD	BRKLNK		BREAK ALL LINKS EXECUTE CODE FOR LINKS
	.WORD	EXELNK		;CHECK FOR SPYING
		SETLNK CLRLNK		;SET LINK FLAGS ;CLEAR LINK FLAGS
	.WURD .WORD	DEFTLK LNKSPC		SET A USER'S TALK STATUS TO DEFAULT SPECIAL FUNCTION HOOK
.IFF	;INC\$LK			
	WORD	RTSR5		SKIP ARGUMENT
	.WORD	RTS PC RTS PC		; NOTHING TO DO ; NOTHING TO DO
	.WORD	RTSPC		; NOTHING TO DO
	.WORD	RTSR5 RTSR5		;SKIP ARGUMENT ;SKIP ARGUMENT
	.WORD	RTS PC PRVERR		; NOTHING TO DO ; PROTECTION VIOLATION ERROR
. ENDC	;INC\$LK			
.IF	NE	INC\$CH		
	.WORD	INHOOK		;INPUT CHARACTER HANDLER
.IFF	;INC\$CH			
		TTSAVE		MAKE THIS A NORMAL CHARACTER
	; INC\$CH	IISAVE		FILLE THE THE THE CHARTERE
. ENDC				
GLOBAL	<rtspc, ORG</rtspc, 	TTSAVE> TTDVRB		;THIS GOES IN THE PAR5 PART
.IF	NE	INC\$CE		
;+				
; SETFU			CTION KE	
;		JUMPED T Y IS PR3		USER LEVEL WRITE SERVICE
-				
SETFUN:	·SPLC	5		UP TO LEVEL 5 FOR SAFETY
DETTON	MOV BNE	TTFUNK(10\$	R1),R4	;;DOES HE HAVE A FUNCTION KEY? ;;YES HE DOES
	BUFFER	GETSML, 10\$,20.	;;GET HIM ONE, LEAVING 20. IN THE POOL ;;EVERYTHING'S OK
	BVC ERROR	NOBUFS		;; NO SMALL BUFFERS FREE
10\$:	TST BEQ	(R3) 40\$;;ARE WE DONE? ;;YES, SE Y'A LATER ;;GET A CHARACTER FROM THE BUFFER
	GETUSR BIC	* C<177		:: AND MAKE IT 7-BIT ASCII
	CMP BEQ	R2, 177 30\$;;IS IT A RUBOUT? ;;YES, IGNORE IT.
	CMP BHIS	R2,#40 20\$;;IS IT CONTROL? ;;NO, LET IT PASS. ;;IS IT LINEFEED?
	CMP BEQ	R2,#12 20\$;;YES, THAT'S OK ALSO
	CMP BNE	R2,#15 30\$;;CARRIAGE RETURN? ;;ALL OTHER CONTROL'S ARE IGNORED
	DIE			

20\$		MOVB	R2,(R4)+	;;STORE IT IN THE FUNCTION KEY
		BIT BNE	#37,R4 30\$;;AT END OF BUFFER? ;;NO, WE'VE GOT MORE TO DO
		SUB BR	#40,R4 60\$;; POINT TO START OF BUFFER ;; AND SAVE IT
30\$;	DEC BR	(R3) 10\$;;ONE LESS CHARACTER TO DO ;;AND DO IT
40\$:	BIT BNE	#37,R4 50\$;;STILL AT START OF BUFFER? ;;NO, SOMETHING IS THERE
		CLR BUFFER	TTFUNK(R1) RETSML	;;HE DOESN'T HAVE A FUNCTION KEY ;;RETURN IT TO THE MONITOR
50\$		BR CLRB	70\$ (R4)	;;AND EXIT ;;END THE COMMAND WITH A NULL
60\$		BIC MOV	#37,R4 R4,TTFUNK(R1)	;;AND POINT TO THE START ;;ENABLE THE KEY
70\$		SPLC	3	;;BACK TO LEVEL 3 ;AND RETURN
GLO	BAL	<ttfunk:< td=""><td>,</td><td></td></ttfunk:<>	,	
.EN	DC	;INC\$CE		
.IF		NE	INC\$LK	
;+ ; S	PYING	G - CHECH	TO SEE IF A USE	ER IS SPYING
;		R1 -> DI	DB	
;		PRIORITY	IS PR5	
;		CALL	SPYING (@SPYING	FROM TTDVR)
;			VILL BE SET IF US	SER NOT LINKED,
;		OR IF US	SER IS SPYING.	
;-				
SPY	ING:	BEQ	10\$;; IS THE USER LINKED? ;; NOPE, READY TO SET THE C BIT
		BIT BEQ	20\$	TTLNKS(R1) ;;SPYING OR ALREADY DOING LINKS? ;;NOPE, DON'T RETURN WITH CARRY SET ;;SET THE CARRY FLAG FOR FAILURE
10\$ 20\$		SEC RETURN		;;AND RETURN (TST CLEARS CARRY)
	BAL	<ttlink, RUBOUTS</ttlink, 	TTLNKS>	
.50	TTL	TMPORG	TTOUPP	RUBOUT CODE
;+		IMPORG	TIDVKK	, NUBOUT CODE
; T	I \$RO	- RUBOUS	rs.	
TIS	RO:	BIT	*TAPE, DDFLAG(R1)	::TAPE MODE?
		BNE TSTB	80\$ TTESCC(R1)	;;YES, IGNORE RUBOUT ;;INTO AN ESCAPE SEQUENCE?
		BNE CALLX	80\$ TECOCX	;;YES, IGNORE A RUBOUT ::CHECK FOR DDT SUB-MODE
		BIT BNE	<pre>#TT2741!NOECHO,I 70\$</pre>	DDFLAG(R1) ;;2741 AND/OR CONFIDENTIAL MATERIAL? ;;YEP, HANDLE SPECIAL
		CALLX BCS	TTGET 60\$;;REMOVE THE PREVIOUS CHARACTER (C=1 IF NONE) ;;NOTHING TO REMOVE
		BITB BGT	<pre>*TTTECS!TTTECO, 100\$</pre>	TIMODE(R1) ;;TECO OR TECO SCOPE MODE(S)? ;;NORMAL TECO, ECHO DELETED CHARACTER
.AS	SUME	TTTECS CMP	EQ 200 R2, #4 0	;;NORMAL GRAPHIC CHARACTER?
		BHIS TSTB	TTMODE(R1)	;;YES, <bs><paint><bs> WILL DO IT ;;TECO SCOPE MODE?</bs></paint></bs>
		BMI TTTECS	50\$ EQ 200	;;YES, LET TECO HANDLE THE 'ECHO'
10\$:	EQ	EKOCTL	
.11		MOV	#040,R5	;; <paint> IS <space></space></paint>
.IF	F			,,,
		MOVB	EKOPNT(R1),R5	;;GET <paint> FROM DDB</paint>
		BIC		;; AND TRIM OFF ANY PARITY
GLO	BAL	<ekopnt:< td=""><td>></td><td></td></ekopnt:<>	>	
.EN	DC	; EKOCTL		
		TSTB BMI	TTMODE(R1) 25\$;;IN TECO SCOPE MODE? ;;YES, ALWAYS DO A SCOPE RUBOUT
. AS	SUME	TTTECS BIT		R1) ;; IN RUBOUT MODE?
		BNE BIS	20\$ #RUBOUT, DDFLAG(I	;;YES R1) ;;NOW HE'S IN RUBOUT MODE
		MOV CALL	#20\$,-(SP) EXELNK	; PRESET RETURN ADDRESS ; AND EXECUTE THIS FOR LINKS
		BIT BNE CALLX	80\$	R1) ;; IS THIS A SCOPE? ;; YES, NOTHING TO DO
		RETURN	ASCOUL, RS, IIBACI	X ;;NO, SO OUTPUT A <\> ;;LOOP FOR LINKED TERMINALS
GLO 20\$		<ttback< td=""><td>EXELNK</td><td>;;EXECUTE THIS FOR LINKS</td></ttback<>	EXELNK	;;EXECUTE THIS FOR LINKS
204		BIT BEQ	<pre>#TTSCOP, TTCHAR() 100\$</pre>	
		CMP BHIS	R2,#40 25\$;;PRINTABLE CHARACTER? ;;YES IT IS
		BIT BEQ		R1) ;;DOES HE HAVE UPARROW SET? 1;NOPE, NOTHING TO DO
25\$		MOV	PC,-(SP) 30\$;;DO THIS TWICE ;;OUTPUT A BACKSPACE
		MOV CALLX	R5,R2 CHOUTE	;;GET PAINT CHARACTER ;;AND OUTPUT IT
30\$	•	CMPB BNE	DDHORZ(R1),#1 40\$; ARE WE "AFTER" THE LAST COLUMN OF THE CRT? ; NO, NOT THIS TIME
		DECB BR	DDHORZ(R1) 80\$	77YES, SO DO A 'FAKE' BACKSPACE
		CALLX	RSXCHR, R5,010 80\$;;OUTPUT A BACKSPACE ;;ALL DONE ;;(RE-)STORE THE CHARACTER WE REMOVED
40\$				
40\$ 50\$ 60\$		CALLX	STOREX #177,R2	;;RESTORE CHARACTER AS <rubout></rubout>
50\$		CALLX MOV CALLX CALLX	#177,R2 TECOCS ENDRUB	;;CHECK FOR SPECIAL MODE(S) ;;END RUBOUT MODE
50\$		CALLX MOV CALLX CALLX CALLX CALL BIT	#177,R2 TECOCS ENDRUB EXELNK	;;CHECK FOR SPECIAL MODE(S) ;;END RUBOUT MODE ;;EXECUTE THIS FOR LINKS R1) ;;SCOPE TERMINAL?
50\$		CALLX MOV CALLX CALLX CALLX	<pre>#177,R2 TECOCS ENDRUB EXELNK #TTSCOP,TTCHAR(I)</pre>	;;CHECK FOR SPECIAL MODE(S) ;;END RUBOUT MODE ;;EXECUTE THIS FOR LINKS
50\$:	CALLX MOV CALLX CALLX CALL BIT BNE	<pre>#177,R2 TECOCS ENDRUB EXELNK #TTSCOP,TTCHAR(1 80\$</pre>	<pre>;;CHECK FOR SPECIAL MODE(S) ;;END RUBOUT MODE ;;EXECUTE THIS FOR LINKS R1) ;:SCOEF TERMINAL? ;;YES, SILENCE IS SILVER ;;OUTPUT A <cr lp=""> ;:CHECK FOR SPECIAL MODE(S)</cr></pre>
50\$ 60\$:	CALLX MOV CALLX CALLX CALL BIT BNE CALLRX CALLX	<pre>#177,R2 TECOCS ENDRUB EXELNK #TTSCOP,TTCHAR(I 80\$ GOCRLF TECOCS</pre>	;;CHECK FOR SPECIAL MODE(S) ;END RUBOUT MODE ;;EXECUTE THIS FOR LINKS R1) ;:SCOPE TERMINAL? ;YTES, SILENCE IS SILVER ;;OUTPUT A <cr lf=""></cr>

#TT2741,DDFLAG(R1) ;;2741 TYPE TERMINAL? (Z-BIT=0 IF SO)
90\$;;DO <BS> CORRECTION (C=Z=0 => RUBBED 2741) BH1 . ENDC :KB2741 80\$: RETURN ;;ELSE JUST EXIT 905: JMPX TTCKBS ;;GO AND DO BACKSPACE CORRECTION ;;GO AND OUTPUT A CHARACTER 100\$: JMPX CHOUTE UNORG . ENDC ;INC\$LK NE INCȘCH .IF .SBTTL HOOK FOR OUR NEW INPUT FUNCTION KEYS ; INHOOK - HOOK FOR THE NEW INPUT FUNCTION KEYS (SP) HAS THE FUNCTION INDEX 0 = CONTROL/E 2 = CONTROL/F 4 = CONTROL/W 6 = CONTROL/X 10 = CONTROL/Y INHOOK::MOV (SP),R3 ;; R3 HAS THE FUNCTION CODE (DON'T POP STACK!) EKOCTL .IF NE BEQ 10\$;;FUNCTION 0 WORKS WITH ECHO CONTROL .ASSUME IN.CE EQ 0 R3,#IN.CW 10\$ CHKINC ;;WAS THE KEY A CONTROL/W ;;YES, THAT WORKS WITH ECHO CONTROL ALSO. ;;SEE IF INPUT FUNCTIONS ARE ALLOWED ;;AND JOIN UP WITH COMMON CODE ;;CHECK TO SEE IF THIS FUNCTION IS ALLOWED CMP BEQ CALLX CALLX CHKNEK 10\$: ; EKOCTL . IFF CALLX CHKINC ::SEE IF INPUT FUNCTIONS ARE ALLOWED - ENDC : EKOCTL @#KISAR6,(SP) ;;SAVE THE CURRENT PAR6 MAPPING (RE-USE STACK) @#KISAR5,@#KISAR6 ;;DUPLICATE OUR PAR5 VALUE #200,@#KISAR6 ;;NOW MAKE IT 4K HIGHER @TISDSP(R3) ;;OFF TO THE ROUTINE (SP)+,@#KISAR6 ;;RESTORE THE KISAR6 MAPPING TTIN04 ;;AND GO TO THE "NORMAL" EXIT MOV MOV ADD 205: CALL JMPX INCOMP - DUMP AN INPUT CHARACTER AND FIX PAR6 THE STACK CONTAINS A RETURN TO THE NORMAL COMPLETION ROUTINE, AND THE SAVED PARG VALUE. THIS ROUTINE WILL POP THE STACK, RESTORE PARG, AND JUMP TO "TTINST", WHICH WILL MAKE THE CHARACTER A "NORMAL" CHARACTER. (SP)+ ;;DUMP THE NORMAL RETURN ADDRESS (SP)+, 0 KISAR6 ;;RESTORE PAR6 TTINST ;;AND PROCESS THE CHARACTER INCOMP::TST MOV JMPX .ENDC ;INC\$CH NE INC\$CE .SBTTL 'E HANDLER ;+ ; TI\$CE- ^E HANDLER ;;THIS GOES IN PAR6 TMPORG TTMAPX TTMAPX ;; THIS GOES IN PAR6 #TTECO!TTECS, TTMODE (R1) ;; TECO/TECO SCOPE MODE? TTNRML ;; MAKE THIS A NORMAL CHARACTER #TDDT, DDFLAG(R1) ;; DDT MODE? TTNRML ;; YES, MAKE THIS NORMAL TTFUNK(R1),R4 ;; GET THE FUNCTION KEY POINTER ZOS ;; NONE, IGNORE THIS CONTROL/E TTESCC(R1) ;; ARE WE IN AN ESCAPE SEQUENCE 205 ;; YES, ANOTHER REASON TO IGNORE THIS (R4)+,R2 ;; GET A CHARACTER ZOS ;; SAVE BUFFER POINTER TTINO9 ;; FORCE" THE CHARACTER TTINO9 ;; FORCE" THE CHARACTER (SP)+,R4 ;; RESTORE BUFFER POINTER #<40-1>,R4 ;; RESTORE BUFFER POINTER #; YES, GO AND GET IT ;; WE'RE DONE... TI SCE : : BIT BNE BIT BNE MOV BEQ TSTB BNE MOVB 10\$: BEQ MOV CALLX MOV BIT 20\$: RETURN GLOBAL (TTFUNK) ENDC : INCSCE .IF NE INCSCH ; MAKE THE INPUT CHARACTER NORMAL TTNRML:: JMP INCOMP UNORG .ENDC ;INC\$CH NE INC\$CY .SBTTL CONTROL/Y HANDLER (DETACH JOB) ; TISCY - DETACH JOB TMPORG TTMAPX .ENABL LSB DDJBNO(R1),R4 ;;GET OWNING JOB # *2 *^C<63.*2>,R4 ;;AND ENSURE A VALID JOB # *2 20S ;;NO JOB, NO DETACH #DDCONS,DDCNT(R1) ;;IS THIS THE CONSOLE? 20S ;;NO, DON'T DETACH JOBTBL(R4),R4 ;;GET JOB DATA BLOCK POINTER 20S ;;NO JOB, NO DETACH #JFNOR,JDFLG(R4) ;;IS THIS LOGGED IN? 20S ;;NOPE, DON'T ACT FOOLISH #JFPRIV,JDFLG(R4) ;;PRIVILEGED USER? TI\$CY:: MOVB BIC BEQ BIT BEQ MOV BEQ BIT BNE BIT

PRCY 10\$: 20\$:	BEQ == CALLX CALLX RETURN	ASCOUT, R5, DETMSC	;;NOPE, IGNORE THIS TH** MAKE NOP TO MAKE CTRL/Y NON-PRIV'D ;;JOUTPUT THE MESSAGE ;JOETACH THE JOB ;;AND THAT'S IT
GLOBAL	< JOBTBL	>	
.DSABL	LSB		
.ENDC	UNORG ;INC\$CY NE	INC\$CX	
.SBTTL	X HAND	LER	
;+ ; TI\$CX		L ALL TYPE-AHEAD	
;-			
	TMPORG	TTMAPX	
TI \$CX : :	CALLRX		;;OUTPUT ^X <cr><lf> ;;AND CANCEL TYPE AHEAD</lf></cr>
	UNORG		
.ENDC	;INC\$CX NE	INC\$CY	
.SBTTL	W HAND	LER	
;+ ; TI\$CW	- ERASE	A WORD	
	TMPORG	TTMAPX	
mt Coll			MADE MODE2
TI\$CW::	BIT BNE TSTB BNE BIT BNE BITB BNE CALLX BCS	<pre>#TAPE,DDFLAG(R1) 10\$ #TTESCC(R1) 105 #TTDDT,DDFLAG(R1) TTNRML #TTECO,TTMODE(I) TTNRML TTCET 105</pre>	;;YES, IGNORE THIS ;;IN AN ESCAPE SEQUENCE? ;;YES, IGNORE THIS `W
	BIT	#NOECHO, DDFLAG (1	R1) ;;SHOULD WE PRINT ANYTHING?
	BNE MOV	20\$ #20\$,-(SP)	;;NO WAY ;;PUSH RETURN ADDRESS FOR LINKS
.IF	NE	INC\$LK	
	CALL	EXELNK	;; EXECUTE THE FOLLOWING FOR LINKS
.ENDC	;INC\$LK		
	BIT	TTSCOP, TTCHAR (R1) ;;SCOPE TERMINAL?
	BNE CALLX	10\$ RSXCHR,R5,'_	;;YES, DON'T OUTPUT AN UNDERLINE ;;OUTPUT AN UNDERLINE
10\$:	RETURN	R2,#11	; AND LOOP FOR LINKS AND CONTINUE ; WAS THE DELETED CHARACTER A TAB?
20\$:	BEQ	60\$; YES, SPECIAL CHECKING NEEDED ; WAS THE CHARACTER A SPACE
	CMP BEQ	R2,#40 60\$; THAT NEEDS SPECIAL CHECKING ALSO
30\$:	BIS BIT	<pre>#RUBOUT, DDFLAG() #NOECHO, DDFLAG()</pre>	R1) ;;INDICATE THAT WE ARE DELETING CHARACTERS R1) ;;SHOULD WE OUTPUT ANYTHING?
	BNE CMP	50\$ R2,#40	;:NO, JUST DELETE MORE- ;:WAS DELETED CHARACTER PRINTABLE?
	BHIS	40\$;;YES, ONE DELETE WILL DO
	BIT BEQ	50\$	R1) ;;WAS UPARROW SET? ;;NO, NO DELETES NEEDED
40\$:	CALL	90\$ 90\$;;DELETE ONE CHARACTER ;;AND ANOTHER
50\$:	CALLX BCC	TTGET 20\$;;GET ANOTHER CHARACTER ;;PROCESS IT IF PRESENT
	BR	70\$;;ELSE CLEAN UP SHOP
60\$:	BIT BEQ	30\$	R1) ;;WERE WE DELETING CHARACTERS? ;;NO, GO BACK FOR MORE
70\$:	CALLX BIC RETURN	STOREX #RUBOUT,DDFLAG(1	;;REPLACE DELETED CHARACTER R1) ;;RESET RUBOUT MODE ;;AND WE'RE DONE
90\$:			
.Ir	EQ	EKOCTL	
	MOV	#40,R2	;;USE <space> FOR PAINT</space>
.IFF			
	MOVB BIC		;;GET PAINT CHARACTER ;;TRIM PARITY
GLOBAL	< EKOPNT	'>	
.ENDC	; EKOCTI		
1005:			



page 102

December 1981

RSTSPROFESSIONAL

.IF	NE	INC\$LK	
	CALL	EXELNK	;;EXECUTE THIS FOR LINKS
. ENDC	;INC\$LK		
	BIT		R1) ;;SCOPE TERMINAL?
	BEQ CALLX	10\$ RSXCHR,R5,010	;;NOPE ;;OUTPUT A BACKSPACE
120\$:	CALLX	CHOUTE RSXCHR, R5,010	;;OUTPUT CHARACTER ;;AND ANOTHER BACKSPACE
	RETURN UNORG		;;LOOP FOR LINKS
. ENDC	;INC\$CW		
.IF	NE	INC\$CF	BY DO DICK NY
.SBTTL	HANDLE	CONTROL/F (OPEN	FILES) DISPLAY
; TISCF	- HANDL	E CONTROL/F	
	TMPORG	TTMAPX	
TI\$CF:	CALL	TTOPNE	;;DO THE OPEN FILES DISPLAY
	BCS RETURN	TTNRML	;;NOT ALLOWED, MAKE CHARACTER NORMAL ;;THAT'S ALL
	TMPORG	TTOPNF	
TTOPNF:			;;NOTHING TO DO UNLESS OVERLAID
	RETURN		;;BYE-BYE
. ENDC	UNORG ; INC\$CF		
.IF	NE	INC\$LK	
.SBTTL	MASTER	ROUTINE FOR TERM	NINAL LINKS
;+	EXELNK	- EXECUTE THE FO	DLLOWING CODE FOR ALL LINKED TERMINALS
;	R1 -> D		
;	CALL EX		
;			LINKED, OR LINKS ARE NOT APPLICABLE,
;	IS LINK	ED, THE ROUTINE	N WITH NO EFFECT. IF THIS TERMINAL FOLLOWING THE CALL WILL BE EXECUTED
;	CALLER	WILL BE EXECUTED	, AND THEN A RETURN TO THE CALLER'S
;	R0 = LI $R1 \rightarrow D$	NE NUMBER TIMES	TWO
;	KI > D	00	
EXELNK:	:CALL	SPYING	;;SHOULD WE DO ANYTHING?
	BCS MOV	20\$;;NOPE, EXELNK IS NOT VALID ;;SAVE OLD R0
	MOV	R1,-(SP) TTLINK(R1),R1	;;AND OLD R1 ;;GET POINTER TO TABLE.
10\$:	MOV	(R1)+,R0	;; AND PUSH THE BYTE COUNT. ;; FICKUP A KEYBOARD NUMBER
	ASL MOV MOV	R0 R1,-(SP)	;;MAKE IT KEYBOARD * 2 ;;SAVE POINTER ;;PICK UP DDB ADDRESS
	BIS	DEV.KB(R0),R1 #LS.LNK,TTLNKS(R2,-(SP)	<pre>;;FICK OF DDB ADDRESS R1) ;;INDICATE THAT WE ARE NOW DOING LINKS ;;SAVE R2</pre>
	MOV	R2,-(SP) R3,-(SP) R4,-(SP)	ISAVE R3
	MOV	R5,-(SP) @20(SP)	;;SAVE R5 ;;AND CALL THE ROUTINE
	MOV	(SP)+,R5 (SP)+,R4	; RESTORE R5 ; RESTORE R4
	MOV	(SP)+,R3 (SP)+,R2	; RESTORE R3 ; RESTORE R2
	BIC	(SP)+,R1	R1) ;;WE ARE NO LONGER DOING LINKS ;;RESTORE POINTER TO TABLE
	DECB	(SP) 10\$; DECREMENT BYTE COUNT ; IF MORE EXISTS, GET IT
	TST MOV MOV	(SP)+ (SP)+,R1 (SP)+,R0	;;JUNK THE BYTE COUNT ;;RESTORE OLD R1 ;;AND OLD R0
20\$:	TST RETURN	(SP) + (SP) +	;;DUMP OUR CALLER'S ADDRESS ;;AND EXIT
GLOBAL	<dev.kb< td=""><td>, TTLINK, TTLNKS></td><td></td></dev.kb<>	, TTLINK, TTLNKS>	
.SBTTL	UTILITY	ROUTINES FOR TE	RMINAL LINKS
;+ ; DEFTL	K - SET	A USER'S TALK ST	ATUS TO THE DEFAULT
;	R1 -> D		
;	PRIORIT	Y IS PR5	
;	CALL	DEFTLK	
;-			
DEFTLK:	BIT BNE	105	<pre>R1) ;;SHOULD WE KEEP THE LINK STATE THE SAME? ;;YES, HE REQUESTED .STAY</pre>
DFTK	MOV	<pre>#LS.QRY,TTLNKS(4 ; **PAT</pre>	R1) ;;SET DEFAULT TALK STATUS CH** DEFAULT TALK STATUS
10\$:	RETURN		;;THAT'S ALL
++	<ttlnks< td=""><td></td><td></td></ttlnks<>		
; LNKSP			LER FOR THE TERMINAL LINK FUNCTIONS
;	R1 -> D		
;	R3 -> X	PECIAL FUNCTION RB ALLING JOB NUMBE	
;	R5 -> (MAPPED) XRB	
;			
;	RETURN		
;	-OR-		
	ERROR	CODE	

; LEGAL	FUNCTIO	NS ARE:	
;	10000	SET A TERMINAL BREAK ALL A TER	LINK
7	10002	SET TALK STATUS	
;	10003	READ TALK STATU	5
;-			
LNKSPC:	:BIC CMP	#10000,R2 R2,#3	;MAKE THE FUNCTION CODE 0 TO 3 ;IS THIS VALID?
	BHI ASL	10\$ R2	;NOPE, ERROR ;R2 HAS FUNCTION CODE TIMES TWO
	JMP	@30\$(R2)	DISPATCH TO THE ROUTINE
10\$:	ERROR	PRVIOL	;INVALID FUNCTION CODE
20\$:	CALLX RETURN	MAPPED, R5, CRELN	K ;CREATE A LINK ;AND WE'RE DONE
30\$:	.WORD	205	CREATE A LINK (MAPPED)
500.	.WURD	BRKSPC	;BREAK A LINK (MAPPED) ;SET TALK STATUS
	.WORD .WURD	SETLKS RDLNKS	;READ TALK STATUS
		RS FOR CREATE A	
.BSECT	HIGH		ERS FOR .SPEC (PRIVILEGED)
CRL.NW:: CRL.IG:			PRINT WARNINGS TERMINAL SETTINGS
CRL.SP: CRL.NQ:	BLKB	. ;SPY CA	
CRD HIg.	.BLKB	. : RESERV	ED
	.BLKB .BLKB	; RESERV ; RESERV	ED
	.BLKB	. ; RESERV	ED
;+	UNORG		
; CRELNE	- SPECI	AL FUNCTION TO	CREATE A LINK
	THRODE	MMM & D.Y	
	TMPORG		
CRELNK::		CHKDET #JFPRIV!JFSYS!J	;ENSURE THAT THE USER IS ATTACHED FNOPR,@JOBF ;IS USER PRIV'D?
		10\$ 1(R3)	;YES, PRESERVE PARAMETERS ;NOT PRIV'D, CLEAR PARAMETER
10\$:	BIT		R1) ;IS TERMINAL SPYING? ;YES, SO ERROR
	BIT	<pre>#CRL.SP, (R3)</pre>	DOES HE WANT TO SPY?
	TST	TTLINK(R1)	; IS HE LINKED?
20\$:	BEQ ERROR	INUSE	;ERROR - CAN'T LINK AND SPY AT THE SAME TIME
	MOV	R1,-(SP)	;SAVE DDB POINTER ;GET KEYBOARD TO LINK WITH
	CMP BLO	R2, #CNT.KB	;IS IT LEGAL? ;YES
	ERROR	NODEVC	KEYBOARD NUMBER IS OUT OF RANGE MAKE IT KEYBOARD * 2
40\$:	CMP	R2 R0,R2	;SAME TERMINAL?
	BNE ERROR	50\$ BADFUO	;NO, THE PARAMETER IS OK ;ERROR - CAN'T LINK TO YOURSELF
50\$:	SPLC	DEV.KB(R2),R2 5	;R2 -> DDB TO LINK TO ;;GO TO LEVEL 5 FOR SAFETY
	BITB BNE	<pre>#1,DDJBNO(R2) 60\$</pre>	;;IS TERMINAL DISABLED? ;;YES, SO DON'T LINK
	TSTB	DDJBNO(R2) 60\$;;IS TERMINAL IN USE? ;;NO, SO HE CAN'T LINK
	BIT	#DDCONS, DDCNT (R	A TO THE THE CONSOLE?
	BNE ERROR	70\$ NOTAVL	;;YES, IT SEEMS OK TO LINK TO ;;ERROR - CAN ONLY LINK TO A JOB'S CONSOLE
	BIT BNE	905	R2) ;;TERMINAL SPYING? ::YES, FAKE THAT HIS TALK IS OFF
	CMP BNE	TTLINK(R1), TTLIN 805	NK(R2) ;; ARE THEY ALREADY LINKED?
	TST	TTLINK(R1) 805	; POSSIBLY, OR ARE THEY BOTH NOT LINKED? ; NIETHER OF THEM ARE LINKED
	BEQ ERROR		; ERROR - THEY ARE ALREADY LINKED
GLOBAL	<dev.kb,< td=""><td>TTLINK, TTLNKS></td><td></td></dev.kb,<>	TTLINK, TTLNKS>	
80\$:	BIT BNE	100\$	<pre>(R3) ;;IGNORING STATUS OR SPYING? ;;YES, SKIP CHECKS</pre>
	BIT BNE	<pre>#LS.RCV, TTLNKS() 100\$</pre>	R2) ;;LINKS ALLOWED? ;;YES, DON'T SAY ANYTHING
	BIT BNE	#CRL.NQ, (R3) 90\$;;IS IT OK FOR US TO WARN HIM? ;;NO, JUST GIVE AN ERROR
	BIT	#LS.QRY, TTLNKS()	R2) ;;DOES USER WANT TO BE WARNED? ;;NOPE.
	B EQ MOVB	DDUNT(R1),R5	; PICKUP TERMINAL NUMBER ; AND GET DDB OF TERMINAL TO NOTIFY
		R2,R1 DDUNT(R1),R0	;; PICKUP KEYBOARD NUMBER TO NOTIFY
	ASL CALLX	R0 ASCOUT, R5, CRLF.	;;AND MAKE IT * 2 0 ;;OUTPUT A <cr lf=""></cr>
	CALL	210\$ ASCOUT, R5, TRYLN	;;AND THE KEYBOARD NUMBER K ;;AND SAY THAT SOMEONE CALLED
90\$:	CALLX ERROR	CHKLIN PRVIOL	;;START UP THE LINE ;;ERROR - TERMINAL NOT RECEIVING LINKS
100\$:	MOV	TTLINK(R1),R4 130\$;; IS OUR CALLER LINKED? ;;YES
	MOV	TTLINK(R2),R4 1205	; IS OUR DESTINATION LINKED?
	BUFFER	GETSML,,20.	;;GET A BUFFER LEAVING 20. IN THE POOL
		110\$ NOBUFS	;;WE HAVE ONE ;;ERROR - OUT OF SMALL BUFFERS
110\$:		R4,TTLINK(R1) R4,TTLINK(R2)	; FOINT TO BUFFER ; IN BOTH DDBS
	ADD	#LT.NXT,R4 LNKLST,R5	;; POINT TO LINK TO NEXT ;;GET FIRST LINK TABLE
	BEQ MOV	115\$ R5,(R4)	;;NONE THERE ;;OUR LINK TO NEXT IS THE CURRENT FIRST LINK
	TST	(R5) +	;;R5 -> LT.PRV OF OLD LINK
. ASSUME	MOV	EQ LT.NXT+ R4,(R5)	LINK THE NEW TABLE TO THE OLD ONE @ LT. PRV
115\$:	MOV TST	R4,LNKLST (R4)+	;;WE ARE NOW THE TOP OF THE LINKED LIST ;;R4 -> LT.PRV OF NEW LINK
.ASSUME		EQ LT.NXT+ #LNKLST,(R4)	2
	BIC	#37,R4 #2,(R4)+	;; POINT BACK TO THE BUFFER'S START ;; INITIALLY, TWO ENTRIES EXIST
	MOVB	DDUNT(R1),(R4)+	;;LOAD KEYBOARD NUMBERS
1000	BR	180\$;; INTO THE TABLE ;; AND EXIT
120\$:	MOV MOV	R2,R0 R1,R2	;;SAVE R2 ;;R2 -> DDB WITHOUT LINK ;;R1 -> DDB WITH LINK
	MOV BR	R0,R1 150\$;;R1 -> DDB WITH LINK ;;NOW PROCESS THIS INSANITY
GLOBAL	(LNKLST.	CRLF.0, TTLINK, T	TLNKS>

GLOBAL <LNKLST, CRLF.0, TTLINK, TTLNKS>

December 1981

130\$:	MOV	TTLINK(R2),R0	;;ARE BOTH TERMINALS LINKED?
	BEO	150\$; NO, REAL CRAZINESS DOESN'T HAPPEN
	MOVB	(R4),R5	; GET USER COUNT FROM OTHER USERS LINK TABLE
	MOV	R4, R2	; COPY OTHER USER'S LINK TABLE POINTER
	TST	(R2) +	: ADVANCE TO FIRST ENTRY IN TABLE
	ADD	R5, R2	; POINT R2 PAST LAST ENTRY IN TABLE
	ADD	(R0),R5	; R5 NOW HAS TOTAL NUMBER OF LINKED TERMINALS
	CMPB	R5, #26.	; MORE THAN 26. TERMINALS?
	BHI	160\$; YES, OH LORD!
	MOVB	R5, (R4)	;;UPDATE SIZE OF NEW LINK TABLE
	MOVB	(R0),R5	; GET OLD LENGTH
	MOVE	R0,-(SP)	SAVE POINTER
	TST	(R0) + (SP)	;;AND POINT TO START
140\$:	MOVB	(R0) + (R2) +	; COPY ENTRY
1409.	MOVB		; AND PICK IT UP
		(R0)+,R1	
	ASL MOV	R1 DEV.KB(R1),R1	;;MAKING IT KB * 2
			;;R1 -> DDB
	MOV	R4, TTLINK(R1)	;; POINT DDB TO TABLE
	SOB	R5,140\$;; AND LOOP
	MOV	(SP)+,R4	;;GET POINTER
	CALL	UNLINK	;;UNLINK THIS BUFFER AND RETURN IT (PAR6)
	BR	180\$;;END OF THIS MADNESS!
150\$:	CMPB	(R4),#26.	;;ARE WE FULL?
	BNE	170\$;;NO, LET HIM ENTER THE TABLE
160\$:	ERROR	NOROOM	;;ERROR - LINK TABLE FULL
170\$:	MOV	R4, TTLINK(R2)	;; PUT POINTER IN DDB
	MOVB	(R4),R0	;;GET SIZE
	INC	(R4)+	;; POINT TO FIRST AND SET SIZE
	ADD	R0,R4	;; POINT AFTER LAST
	MOVB	DDUNT(R2), (R4)	;;AND STORE THIS
180\$:	MOV	(SP)+,R1	;;RESTORE DDB POINTER
	BIT	#CRL.SP, (R3)	;;DOES HE WANT TO SPY?
	BEQ	190\$;;NO, DON'T PLAY WITH THE DDB
	BIS	#LS.SPY, TTLNKS ()	R1) ;;THIS TERMINAL IS NOW SPYING
	BR	220\$;;SKIP ANNOUNCEMENT
GLOBAL		, TTLINK, TTLNKS>	
190\$:	BIT	<pre>#CRL.NW,(R3)</pre>	;;SHOULD WE NOTIFY?
	BNE	220\$;;NOPE
	MOVB	DDUNT(R1),R5	;;GET KEYBOARD NUMBER
	MOV	R5,R0	;;ALSO INTO RO
	ASL	RO	;;MAKE IT TIMES 2
	CALLX	ASCOUT, R5, ANNLN	K ;; PRINT SUCCESS ANNOUNCEMENT
	CALL	210\$;;PRINT IDENTIFIER
	CALLX	ASCOUT, R5, CRLF.	0 ;;AND DO A <cr lf=""></cr>
	CALLRX	CHKLIN	;;STARTUP LINE
210\$:	CALLX	ASCOUT, R5, ANLNK	1 ;;OUTPUT "KB"
	MOV	R5,R3	;;GET KEYBOARD NUMBER
	CALL	NUMOUT	;;AND OUTPUT IT
	CALLX	ASCOUT, R5, ANLNK	
	MOV	JOBJD2,R2	;;GET SECOND JOB DATA BLOCK
	MOV	J2PPN(R2),R2	;;NOW GET THE PPN
	MOV	R2,R3	;;AND SAVE IN R3
	SWAB	R3	;;GET THE PROJECT CODE
	BIC	#^C<377>,R3	;; AND TRASH THE PROGRAMMER
	CALL	NUMOUT	;;OUTPUT IT
	CALLX	RSXCHR, R5,054	;;OUTPUT A ","
	MOV	R2,R3	;;GET PPN AGAIN
	BIC	# C<377>,R3	; THIS TIME GETTING PROGRAMMER
	CALL	NUMOUT	;;OUTPUT IT
	CALLX	RSXCHR, R5, ']	;;OUTPUT A "]"
220\$:	RETURN		; NOW WE'RE DONE
GLOBAL	<crlf.0< td=""><td>></td><td></td></crlf.0<>	>	

UNORG

; BRKSFC - SPECIAL FUNCTION TO BREAK A LINK

; CHECK IF THE USER IS ATTACHED BRKSPC::CALL CHKDET

SPLC 5 ;;UP TO LEVEL 5 CALLX MAPPED,R5,BRKLNK ;;BREAK THE LINK

RETURN	;;AND OFF

;* ; BRKLNK - BREAK A TERMINAL'S LINKS ; UNLINK - UNLINK A LINK TABLE AND RETURN IT TO THE MONITOR'S POOL R1 -> DDB R4 -> LINK TABLE (FOR UNLINK) CALL MAPPED, R5, BRKLNK

CALL MAPPED, R5, UNLINK

RETURN WITH ALL REGISTERS PRESERVED

R5 = RANDOM (FOR UNLINK)

TMPORG TTMAPX

.ENABL LSB

;; SAVE ALL REGISTERS TTLINK(R1),R4 ;; PICK UP LINK TABLE POINTER 605 ;; NONE, SO DO NOTHING HLS.LNKIES.SPY,TTLINK(R1) ;; CLEAR LINK POINTER (R4),R2 ;; GET LINK COUNT R4,R3 ;; COPY POINTER (R3) +, DUNT(R1) ;; IS THIS OUR ENTRY? 105 ;: NO, LOOP R4,R2 ;; RC -> LAST ENTRY - 1 1(R2),-(R3) ;; UPDATE LINK TABLE (R4) ,1 ;; ONLY ONE USER LET? 205 ;; NO 2(R4),R0 ;; RET UNIT NUMBER BRKLNK: : REGSCR MOV BEQ BIC CLR MOVE MOV 105: CMPB BNE ADD DECB CMPB BNE MOVB ASL MOV BIC CLR (R4),*1 ;;ONLY ONE USER LEFT? 20S ;;NO 2(R4),R0 ;;GET UNIT NUMBER R0 ;;MAKE THIS UNIT NUMBER * 2 DEV.KB(R0),R1 ;;PICK UP DDB ADDRESS TLLINK(R1) ;;AND REMOVE POINTER UNLINK ;;DUMP BUFFER AND EXIT (R4),R2 ;;GET LINK COUNT (R3)+ ;;POINT TO END + 1 -(R3),R0 ;;GET KEYBOARD NUMBER R0 ;;MAKE IT KEYBOARD * 2 DEV.KB(R0),R1 ;;FICK UF NOR DE MITAL? 60S ;;NO, LINK CAN STAY R2,30S ;;LOOP FOR MORE (R4),R2 ;;RE-GET LINK COUNT (R3)+,R0 ;;GET KEYBOARD NUMBER R0 ;;MAKE IT KEYBOARD * 2 DEV.KB(R0),R1 ;;FICK UP DDB ADDRESS * 12,0S ;;LOOP FOR MORE R0 ;;MAKE IT * 2 DEV.KB(R0),R1 ;;FICK UP DDB ADDRESS TLLINK(R1) ;;AND REMOVE POINTER R2,40S ;;LOOP FOR ALL ENTRIES BR 205: MOVE TSTE 30\$: MOVB ASL MOV BIT BEC SOB MOVE MOVE ASL MOV BIC 40\$: CLR



For complete details, contact us at:

PlyCOM services; inc. P.O. Box 160

Plymouth, IN 46563 (219) 935-5121

CIRCLE 42 ON READER CARD

RSTS RESCUE SQUAD

We salvage all kinds of disasters:

- unreadable disks
- immediate response
- on-site
- custom recovery
- repaired telephone DIAL-UP

ruined UFDs and MFDs

- software tools
- 90% success to date
- more than 1 GB rescued to data

Brought to you by **On-Track Systems, Inc.**

and a well known (and read) Disk Directory expert.

> **CALL 24 HOURS** 215-542-7133

page 104

December 1981

RSTSPROFESSIONAL

UNLINK: .ASSUME 50\$:	MOV BEQ TST LT.PRV MOV BUFFER	R5,@LT.PRV(R4) 50\$ (R5)+ E0 LT.NXT+2	;;LINK NEXT TABLE TO PREVIOUS TABLE ;;RETURN BUFFER		
60\$:	RETURN		;;AND EXIT		
GLOBAL		, TTLINK, TTLNKS>			
.DSABL	LSB				
;+ ; SETLK: ;-	UNORG S - SET	A TERMINAL'S TAL	K STATUS		
SETLKS: KSPR 10\$:	CLR BIT BNE MOV == BIC BIT BEQ ERROR BIC	10\$ #LS.STY,R4 2;; **PA? R4,(R3) R4,TTLNKS(R1) 10\$ PRVIOL	;ENSURE ATTACHEDNESS ;INITIALLY NO PRIVILEGED BITS NORPR, @JOBF ;IS CALLER PRIVILEGED? ;PRIVILEGED, SO DON'T MASK FUNCTIONS ;R4 HAS THE PRIVILEGED BITS FCH+* MODES THAT ARE PRIVILEGED ;CLEAR THE PRIVID BITS FOR NON-PRIV'D USERS ;ANY PRIV'D BITS SET NOW? ;NOPE, LET THIS PASS ;SORR (R3) ;CLEAR OUT THE "SYSTEM* BITS ;SET THE TALK STATUS		
;+ ; RDLNK;	MOV RETURN S - READ	(R3), TTLNKS(R1) A TERMINAL'S TAI	;AND EXIT		
, RDLNKS:	:CALL TST MOV BIC RETURN	CHKDET (R5)+ TTLNKS(R1),(R5) #LS.SPY!LS.LNK,	;ENSURE ATTACHEDNESS ;ADVANCE TO XRBC ;RETURN LINK STATUS (R5) ;CLEAR THE SYSTEM STUFF ;BACK TO THE USER		
;+ ; CHKDE' ;-	; CHKDET - CHECK IF A USER IS DETACHED				
CHKDET:	CMPB BNE BIT BEQ RETURN ERROR	DDJBNO(R1), JOB 10\$ #DDCONS, DDCNT(R1 10\$ DETKEY	;IS USER ATTACHED ;NO);YES, BUT IS THIS THE CONSOLE? ;NO IT ISN'T ;EVERYTHING'S COOL ;GIVE HIM AN ERROR		
GLOBAL	<ttlink< td=""><td>,TTLNKS></td><td></td></ttlink<>	,TTLNKS>			
;+ ; CHKLN	K - CHEC	K LINK FLAGS			
; ; ;-	CALL	CHKLNK, R5, <mask< td=""><td>TO TEST></td></mask<>	TO TEST>		
.ENABL	LSB				
CHKLNK:	BIT	10\$ (R5)+,@TTLINK(R] R5	;;LINKS IN USE? 1) ;;DO THE TEST. ;;AND EXIT		
;+ ; SETLN	K - SET	LINK FLAGS			
; ; ; ;-	CALL	SETLNK, R5, <mask< td=""><td>TO SET></td></mask<>	TO SET>		
SETLNK:	BIS	10\$ (R5)+,@TTLINK(R] R5	;;LINKS IN USE? 1) ;;SET THE BITS ;;AND EXIT		

;+ ; CLRLNK - CLEAR LINK FLAGS

RSTS/E USERS
SYS CALL LIBRARY
FOR
RT-II & RSX EMULATORS
\$500/CPU
AVAILABLE 9 TRACK TAPE ONLY CONTACT:
C. MUSUMECI
NATIONAL PUBLIC RADIO
2025 M ST., N.W.
WASHINGTON, D.C. 20036
202-822-2644

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	CALL	CLRLNK, R5, <mask< td=""><td>TO RESET></td></mask<>	TO RESET>
CLRLNK::	CALL	10\$;;LINKS IN USE?
	DIC) ;;CLEAR THE BITS. ;;AND EXIT
10\$:	TST	TTLINK(R1)	;;LINKS IN USE?
	TST	(SP) +	;;YES. ;;NO, DUMP CALLER
	TST SEZ	(85)+	;;SKIP ARGUMENT ;;SET ZERO FLAG FOR FAILURE
	RETURN	R5	;;AND EXIT ;;NORMAL EXIT
	RETURN		FINDREAD DATE
.DSABL	LSB		
GLOBAL	<ttlink></ttlink>	•	
; NUMUUT	NUMUUT - OUTPUT AN INTEGER		
	R1 -> DE		
;	R3 = NUM	BER TO OUTPUT	
;	CALL NUM	OUT	
	TMPORG	THE REAL PROPERTY	THIS GOES IN THE PAR6 PART
NUMOUT::		R5	;;SAVE ALL REGISTERS ;;R5 WILL BE BYTE COUNT
	CLR DIV	R2 #10.,R2	;;CLEAR MSB FOR DIVIDE ;;DIVIDE BY 10.
		#60,R3	;;CONVERT TO ASCII ;;PUSH NUMBER
	INC	R5	; ADJUST BYTE COUNT ; AND SET UP FOR ANOTHER DIVIDE
	BNE	10\$;;LOOP UNLESS WE'RE DONE
		CHOUTE	;;RESTORE DIGIT ;;AND OUTPUT IT
		R5,20\$;;LOOP FOR MORE ;;EXIT (RESTORING REGISTERS)
	UNORG		
	;INC\$LK EQ	INC\$LK	
.SBTTL	SKIP AN	ARGUMENT AND RET	URN
;+			
; RTSR5	- SKIP A	AN ARGUMENT AND R	ETURN
	CALL	RTSR5,R5, <arg></arg>	
;	THIS ROU	JTINE IS NEEDED I	F TERMINAL LINKS ARE NOT GENERATED
	THE Z FI	LAG IS SET ON RET	URN.
;-			
RTSR5:	CMP	(R5),(R5)+	;SKIP ARGUMENT AND SET ZERO FLAG
	RETURN		; AND GO BACK
.ENDC	;INC\$LK NE	INC\$CH	
.SBTTL	INPUT CH	ARACTER JUMP VEC	TORS ARE DEFINED HERE
	TMPORG	TTMAPX	
TISDSP:		E	
	DEFVEC	F W	
	DEFVEC	х	
	DEFVEC	Y	
	.DSECT		;DEFINE OFFSETS
			;CONTROL/E
	.BLKW .BLKW		;CONTROL/F
IN.CW:	.BLKW		; CONTROL/F ; CONTROL/W ; CONTROL/X
IN.CW: IN.CX:	.BLKW		;CONTROL/F
IN.CW: IN.CX: IN.CY:	.BLKW .BLKW .BLKW		; CONTROL/F ; CONTROL/W ; CONTROL/X
IN.CW: IN.CX: IN.CY:	.BLKW .BLKW .BLKW .BLKW UNORG ;INC\$CH		;CONTROL/F ;CONTROL/W ;CONTROL/X ;CONTROL/Y
IN.CW: IN.CX: IN.CY: .ENDC .SBTTL	.BLKW .BLKW .BLKW .BLKW UNORG ;INC\$CH DEFINE (USEFUL ASCII STRI	;CONTROL/F ;CONTROL/W ;CONTROL/X ;CONTROL/Y
IN.CW: IN.CX: IN.CY: .ENDC .SBTTL	.BLKW .BLKW .BLKW .BLKW UNORG ;INC\$CH DEFINE (TMPORG	USEFUL ASCII STRI	;CONTROL/F ;CONTROL/W ;CONTROL/X ;CONTROL/Y
IN.CW: IN.CX: IN.CY: .ENDC .SBTTL .ENABL	.BLKW .BLKW .BLKW .BLKW .BLKW UNORG ;INC\$CH DEFINE (TMPORG LC	USEFUL ASCII STRI TTMAPX	;CONTROL/F ;CONTROL/W ;CONTROL/X ;CONTROL/Y
IN.CW: IN.CX: IN.CY: .ENDC .SBTTL .ENABL .IF	.BLKW .BLKW .BLKW .BLKW UNORG ;INC\$CH DEFINE (TMPORG LC NE	USEFUL ASCII STRI TTMAPX INCSCX	;CONTROL/F ;CONTROL/W ;CONTROL/X ;CONTROL/Y
IN.CW: IN.CX: IN.CY: .ENDC .SBTTL .ENABL .IF TTICXC::	.BLKW .BLKW	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200	;CONTROL/F ;CONTROL/W ;CONTROL/X ;CONTROL/Y
IN.CW: IN.CX: IN.CY: .ENDC .SBTTL .ENABL .IF TTICXC::	.BLKW .BLKW .BLKW .BLKW UNORG ;INC\$CH DEFINE (TMPORG LC NE	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200	;CONTROL/F ;CONTROL/W ;CONTROL/X ;CONTROL/Y
IN.CW: IN.CX: IN.CY: .ENDC .SBTTL .ENABL .IF TTICXC:: .ENDC	.BLKW .BLKW .BLKW .BLKW .BLKW UNORG ;INCSCH DEFINE (TMPORG LC NE :.ASCII ;INCSCX NE	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY	; CONTROL/F ; CONTROL/W ; CONTROL/X ; CONTROL/Y
IN.CW: IN.CX: IN.CY: .ENDC .SBTTL .ENABL .IF TTICXC:: .ENDC	.BLKW .BLKW .BLKW .BLKW .BLKW UNORG ;INCSCH DEFINE (TMPORG LC NE :.ASCII ;INCSCX NE	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY	;CONTROL/F ;CONTROL/W ;CONTROL/X ;CONTROL/Y
IN.CW: IN.CY: IN.CY: .ENDC .SBTTL .ENABL .IF TTICXC:: .ENDC .IF DETMSG:	.BLKW .BLKW .BLKW .BLKW .BLKW UNORG ;INCSCH DEFINE (TMPORG LC NE :.ASCII ;INCSCX NE	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY	; CONTROL/F ; CONTROL/W ; CONTROL/X ; CONTROL/Y
IN.CW: IN.CY: IN.CY: .ENDC .SBTTL .ENABL .IF TTICXC:: .ENDC .IF DETMSG:	BLKW BLKW BLKW BLKW UNORG ;INCSCH DEFINE U TMPORG LC NE :ASCII ;INCSCX NE :ASCII ;INCSCY	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY	; CONTROL/F ; CONTROL/W ; CONTROL/X ; CONTROL/Y
IN.CW: IN.CY: IN.CY: ENDC .SBTTL .ENABL .IF TTICXC:: .ENDC .IF DETMSG:: .IF	BLKW BLKW BLKW BLKW UNORG ;INCSCH DEFINE C TMPORG LC NE :ASCII ;INCSCX NE :ASCII ;INCSCY NE	USEFUL ASCII STRI TTMAFX INCSCX %^X%<15><12><200 INCSCY <15><12>%Detachi	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS
IN.CW: IN.CY: IN.CY: ENDC .SBTTL .ENABL .IF TTICXC:: .ENDC .IF DETMSG:: .IF	BLKW BLKW BLKW BLKW UNORG ;INCSCH DEFINE (TMPORG LC NE ASCII ;INCSCX NE : ASCII ;INCSCY NE : ASCII	USEFUL ASCII STRI TTMAFX INCSCX %^X%<15><12><200 INCSCY <15><12>%Detachi INCSLK	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS
IN.CW: IN.CY: IN.CY: ENDC .BBTTL .ENABL .IF TTICXC:: .ENDC .IF DETMSG: .ENDC .IF ANNLNK:	BLKW BLKW BLKW BLKW BLKW UNORG ;INCSCH DEFINE U TMPORG LC NE ASCII ;INCSCX NE : ASCII ;INCSCY NE : ASCII : ASCII : ASCII	USEPUL ASCII STRI TTMAPX INCSCX %^X%<15><12><200 INC\$CY <15><12>%Detachi INC\$LK <15><12>%Link fr	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS
IN.CW: IN.CY: IN.CY: ENDC .SBTTL .ENABL .IF TTICXC:: .ENDC .IF DETMSG:: .ENDC .IF ANNLNK:: ANLNK:	BLKW BLKW BLKW BLKW BLKW BLKW BLKW BLKW	USEFUL ASCII STRI TTMAPX INCSCX %^X%<15><12><200 INCSCY <15><12>%Detachi INC\$LK <15><12>%Link fr %KB% % [%	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS)> (ng%<15><12><14><200>
IN.CW: IN.CY: IN.CY: ENDC. SBTTL .ENABL .IF TTICXC:: .ENDC .IF DETMSG:: .ENDC .IF ANNLNK:: ANLNKI: TTILK:	BLKW BLKW BLKW BLKW BLKW BLKW BLKW BLKW	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY <15><12>%Detachi INC\$LK <15><12>%Link fr %KB% % [% % attempted to]	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS
IN.CW: IN.CY: IN.CY: IN.CY: ENDC .SBTTL .ENDC .IF TTICXC:: .ENDC .IF ANNLNK:: ANLNKI: .ENDC	BLKW BLKW BLKW BLKW BLKW BLKW BLKW BLKW	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY <15><12>%Detachi INC\$LK <15><12>%Link fr %KB% % [% % attempted to]	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS)> (ng%<15><12><14><200>
IN.CW: IN.CY: IN.CY: IN.CY: ENDC .SBTTL .ENDC .IF TTICXC:: .ENDC .IF ANNLNK:: ANLNKI: TTYLNK: .ENDC	BLKW BLKW BLKW BLKW BLKW BLKW BLKW BLKW	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY <15><12>%Detachi INC\$LK <15><12>%Link fr %KB% % [% % attempted to]	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS)> (ng%<15><12><14><200>
IN.CW: IN.CY: IN.CY: IN.CY: ENDC .SBTTL .ENDC .IF TTICXC:: .ENDC .IF ANNLNK:: ANLNKI: .ENDC	BLKW BLKW BLKW BLKW BLKW BLKW BLKW BLKW	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY <15><12>%Detachi INC\$LK <15><12>%Link fr %KB% % [% % attempted to]	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS)> (ng%<15><12><14><200>
IN.CW: IN.CY: IN.CY: IN.CY: IN.CY: ENDC .SBTTL .ENDC .IF TTICXC:: .ENDC .IF ANLNK:: ANLNK: .ENDC .DSABL	BLKW BLKW BLKW BLKW BLKW BLKW BLKW BLKW	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY <15><12>%Detachi INC\$LK <15><12>%Link fr %KB% % [% % attempted to]	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS)> (ng%<15><12><14><200>
IN.CW: IN.CY: IN.CY: IN.CY: ENDC .SBTTL .ENDC .IF TTICXC:: .ENDC .IF ANNLNK:: ANLNKI: .ENDC	BLKW BLKW BLKW BLKW BLKW BLKW BLKW BLKW	USEFUL ASCII STRI TTMAPX INC\$CX %^X%<15><12><200 INC\$CY <15><12>%Detachi INC\$LK <15><12>%Link fr %KB% % [% % attempted to]	; CONTROL/F ; CONTROL/W ; CONTROL/W ; CONTROL/Y :NGS)> (ng%<15><12><14><200>

December 1981

19

page 105 RSTSPROFESSIONAL 12-Nov-91 20:21 TARAT SYSSEN.CHO *G/TBL/TEL/VKCr>
TBL/TEL/VKCr>
TBL/TEL/C=IN:COMMON/KERNEL/DK:CONFIG/IN:CHECK/TELKCr>
#IKcr> 14 H4CRD_SAVCCP> LINKS/LINKS/CSIN:CJHHON,KERNEL/DK:SONFIG/IN:CHECK/KBDEF/LINKS<cr> Cambo-SG/TDVR/VCCP TIOVA/CCCP> +1(CCP) Cambo-SVCCP> MACRO. SAVSer 12-Nov-81 20:21 14.41 SYSGEF. CHE =G/LINKS,LINKS/V<cr> LINKS,LINKS/C=IN:COMMON,KERNEL,DK:CONFIG,IN:CHECK,KBCEF,LINKS<cr> *AICcr>
IR MACRO_SAVCcr>
TTOPNF/TTOPNF/C=IN:COPMON/KERHEL/DK:TTOPNF/Cer>
LINKS/CCr>
LINKS/CCr>
*AICcr>
TTOPNF/C(cr>
<GBC0>WVCcr>
NI:RBTSCcr>
*ExCcr>
ExCcr> I hope that this monitor extension is useful. If you have any comments, suggestions, or "wishes" for future articles, please drop me a line. A complete source distribution of terminal links is available. To order, please send check or money order to: Software Techniques, Inc., 5242 Katella, Los Alamitos, CA 90720 The order numbers are as follows: \$50 A1080-YD (9-track 800 BPI magtape) \$50 A1080-YM (9-track 1600 BPI magtape) \$20 A1080-YZ (Printed listing only) How to make your DEC computer talk to almost any other computer. Purchase orders will not be accepted. Send the data to our INSTOR/80, where it's written on an 8" diskette in IBM 3741 format. Then read the diskette into the other computer. If your other computer can't read IBM's format, connect your INSTOR/80 to it. Fast and simple. Standard RS-232 interface. Standard diskette media. To get your computers talking to each other, call our sales people at INSTOR. (415) 326-9830. micro's, mini's, data entry and capture systems, and nearly anything using RS-232. MAS 6 175 Jefferson Drive Menlo Park, CA 94025 CIRCLE 85 ON READER CARD D8.8 **RSTS PROFESSIONAL** Name Address Box 361 · Ft. Washington, PA 19034-0361 · (215) 542-7008 □ PAYMENT ENCLOSED for one year's subscription (6 issues). Suite _ US 3rd class, \$35 / Canada & US 1st class, \$50 / All other countries air mail, \$60, payable in US dollars. City/State/Zip _ □ BILL ME for one year's subscription: Country_ Phone (□ US 3rd class / □ Canada or US 1st class / □ Other foreign.)____ FREE CLASSIFIED AD WITH SUBSCRIPTION !! Please send BACK ISSUES circled: Vol. 1, #1 Vol. 2. #3 Vol. 3. #2 □ \$10 per issue enclosed. Vol. 2. #1 Vol. 2. #4 Vol. 3. #3

Vol. 3, #1

ML

Vol. 3, #4

XL

Vol. 2. #2

S

□ Bill me for \$12.50 per issue.

Send me a RSTS PRO

Tee Shirt - \$6.95

Your first 12 words are absolutely FREE, only \$1.00 per word thereafter. Use the space provided below.

V3.4



TSXplus AN ALTERNATIVE TO RSTS

By Ed Judge, Northampton, MA 01060

I am writing this little piece to acquaint the RSTS user with another timesharing operating system that runs on DEC computers, TSX and TSXplus. Both are from S&H Computer, in Nashville, Tennessee. They are quite popular among smaller (11-03,23,34) system owners.

S&H Computers was founded in 1974 by Harry R. Sanders, who is still president. He was a professor of Electrical Engineering at Vanderbilt University, teaching Digital Circuits and related electronics. Phil Sherrod, one of the principal writers of the system, was helping develop an operating system and later a FOR-TRAN compiler for a Xerox Sigma 7 computer. They got together and TSX was written.

TSX, the first available system in this line, was available around 1976. It executed with the RT11 monitor as a program that controlled who was in core and how long he stayed there. It used only 64K of memory and swapped workspaces to a file on disk storage. With hard disks, performance with 2 or 3 terminals was acceptable. It cost \$1200 and had a spooler.

When memory above 64K was available, an XM option was made available that allowed the upper memory portion to be used as a swapping area, and when it was filled, it would swap to disc. Later, a device handler (VM.SYS — Virtual Memory) was created that treated the extra memory like a disk (and is still around in public domain). The swap file was placed there, speeding up response time significantly. These were the hot setups among TSX users until TSX-Plus became available in August 1980.

TSX-Plus is like RSTS, and in most instances, runs as fast or faster than RSTS, but costs only \$2000. It maps user jobs into extended memory and executes them there. It can use all 256K of memory available with the 18 bit address capability of the 11-23 (or whatever) in a very efficient manner. A 22 bit system will be available later when the 22 bit QBUS becomes available from DEC. The small overhead it introduces and its simple but efficient time slice algorithm are the secret of its fast response time. It has a protected environment for each user space, spooling, spooling backup, forms support for the spooler, dynamic alteration of constants in the memory swapping algorithm that allows the priviledged operator to optimize the system to the current workload, logon and system usage accounting. It has a debugger that can be loaded into the system without the need of linking it to the program in question. Interterminal message facilities allow notes to be passed, printed on the terminal at log-on time. Both of these systems show the single user an RT11 environment having a max of 56KB workspace, with certain enhancements mentioned below. They are a few restrictions, as certain commands from the single-user RT11 environment would be unacceptable in a multiuser system.

Terminals can be locked to certain jobs or devices, so that any nonpriviledged users cannot peruse the system. Records can be locked to one job or shared. Many different terminals are supported, and protocols for many are available, which set pagesize, rubout mode, clear screen if appropriate. Xon-Xoff, etc. Multiplexer lines as well as standard serial and parallel lines are supported, along with modems and timeout. More than 15 lines can be connected, the writers claim, although the most I have seen is 6 terminals and 3 printers on an 11-23 with 256KB (with very good response times, I might add). TT buffers, data caches, and the performance monitor are stored in extended memory to preserve useable workspace memory. A user can have a full 56KB of workspace without any system code taking up space. A full-featured BASIC can be loaded and will still leave over 40KB of useable memory. If this space is not needed, the memory limit can be lowered to permit more users in core.

page 107 RSTSPROFESSIONALRSTSPROFESSIONA

Device directory file caching is used to enhance speed on lookups, and an option to cache whatever device directory you would like can further speed up programs. A priviledged user can set this at runtime. A performance monitor can be run to see what the system is doing, and a graph of system memory usage can then be printed out and used to make decisions on how to further improve software performance. It also allows the priviledged user to lock certain time-critical or real-time jobs in core, and can give special jobs direct access to the I/O page. System and time statistics are continuously available, such as I/O CPU, Idle and Swapping times etc. for further optimizing the systems response.

Virtual lines allow a single terminal to control many separate jobs at once, and detached jobs run independent of any terminal. Command files are executed as though they were system commands. They can also pass several parameters to the file about to be executed. When an input is made, the system first looks on the system disc for an image file of that name, passing the listed parameters, if any. It continues to look, first for a command file on the system disk, then a command file on the storage disk, and finally, an image program on the storage disk. This allows you to effectively extend system commands with your own custom set of operations.

Most important, it is easy to set up. Changes, such as adding new terminals and printers etc. can be done quickly, and the system regenerated and brought up in much less than 30 minutes, usually less than 10. This, along with the ability to dynamically change system constants such as the time slice constants and memory partitions at the terminal and watch the results, are what I find particularly useful. Compare this to RSTS or RSX, especially the system generation times.

My system (11-23, 256KB, FP, RL01, RL02, RX02) with an overfull compliment of handlers, buffers, options, etc., occupies about 42KB, leaving about 206KB for program space, responds extremely quickly. I have three terminals and 2 spooling printers, and unless I am running a disk sort, I see no noticeable lag in response time when all are operating. Changing system swapping algorithm constants can make significant differences in response, though the defaults are very good. Playing with them and watching the results on the performance monitor is very enlightening as to which type of programs take up system resources.

There are some utilities and programs that cannot be run under TSX-Plus that run under RT11, noteably those that access the I/O page. When I wish to run these, I shut the system down by not allowing any logons. When the last user is off, the system automatically boots up RT11. After I finish, I boot up TSX-Plus again (a matter of about 10 seconds) and away I go (there are certain advantages to a small system).

TSX and TSX-Plus are trademarks of S&H Computers, Nashville, Tenn.

Cartoon by Kevin Macey, Submitted by Peter Dick, Silver Programs, London



FREE **BASIC PLUS TWO** LICENSE

In the spirit of the holidays, Amcor has something special for those of you who are interested in our comprehensive line of software products, but do not have Basic Plus Two on your system. Through January 31, 1982, Amcor will provide a Basic Plus Two license to you at NO COST (that's free!) if you license any two Amcor products. That's right, FREE, and in the process you will acquire the most advanced application products available for the PDP-11, RSTS world.

Amcor software products are installed around the world and all over the USA. The foundation of Amcor application products is AMBASE, the most comprehensive DBMS/Application Development Tool available for PDP-11, and soon for VAX computers.

If you haven't looked at our software, now is a great time. Just drop us a note, or give us a call TOLL FREE at 1-800-626-6268, we will be happy to answer any of your questions.

Headquarters 1900 Plantside Dr. Louisville, KY 40299, Regional Offices: Nationwide

amcor computer corp

CIRCLE 86 ON READER CARD

RSTSPROFESSIONALRSTSPROFESSIONA

CLASSIFIED

Software Support from Free Lance Logic Santa Ana, CA (714) 979-7860.

RSTS optimization done in Eastern Massachusetts, Call David, 327-9476 (nights).

PROGRAMMING FLUID printed on coffee mug. \$6.50 from VIA OZ, GPO Box 1241, NYC USA 10116.

EFEL SYSTEMS High Quality Truly Interactive Software POB 21501 Tel-Aviv, Israel.

Allied Data offers ERIS which allows you to build an input model to FINAR from your report files; increases accuracy and saves input time. ALLIED DATA (206) 456-3535.

The ultimate RSTS/E performance tool! Call (614) 764-1585.

BASIC-PLUS-2/RMS/COBOL

Contracting assignments wanted by consulting firm in Boston area. Total of 24 years experience. Very competitive rates. 617-646-9656 Tab Data Systems, Inc. 65 James St., Arlington, MA 02174.

Need Digital Software? Create your own with RSTS/E Application Generator. Call Tom Meredith (503) 245-1341.

Give 'em heck, buy DEC. Hello to all at Rose Hulman.

McGrath Consulting Service, 15 yrs. programming & Design experience. RSTS/E Basic + Dibol (415) 449-1428.

RSTS/E PROFESSIONALS

Positions now open in exciting WALL STREET projects.

-TOP MONEY--SUBCONTRACTORS WELCOMED-

Send resume or call: LINK SOFTWARE INC. P.O. Box 208, Wall Street Station, New York, 10268 (212) 426-8921

SAVE YOUR quarters, games for RSTS/E are here. Interactive, real-time games for VT52 and VT100 terminals. Some of the games available:

BLKADE: As many as eight players try to force each other to collide with their growing tails on the screen.

STRWRS: You have five minutes to destroy the Death Star with your single X-wing fighter.

SUBS: Two players manuever around islands in an attempt to sink the other player's submarine.

Games come on 9-track magtape with their own high-quality user manuals. Order the first game for \$39.95, \$29.95 for each additional game. Individual manuals available for \$9.95 each.

Send your check, or write or call:

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



BUYING AND SELLING SYSTEMS • CPU'S • PERIPHERALS • TERMINALS OPTIONS • MEMORY • COMPATIBLES

CALL DICK BAKER (305) 979-2844



Send Classified Ads to: 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.]

BASIC PROGRAMMERS. Well established RSTS time-share service located near San Diego running five 1170s and VAX is looking for programmers/analysts with experience in business/commercial applications. Basic Plus, Basic-2, or Vax-Basic experience required. Send resumes with salary history to: NCCS, 2235 Meyers Avenue, Escondido, CA 92025 (714) 745-6006.

Southwest Automation Complete Computer Services Sales, Lease, Purchase, Repair, Software, Timesharing. (713) 371-9288.

For Sale Xerox Sigma-9. Buy now, get free Mahendra! (306) 584-4632.

Programmer required: RSX-11M, Unibus Handler for DMA Data Cartridge System. PO Box E1425, Binghamton, N.Y. 13902.

RSTS Time available. Bulk or Measured. Dual 11/70's - Minneapolis Location Data Directions Inc. (612) 333-0580.

RSTS/E RT11 Consulting. Basic +, DIBOL, reasonable rates. Horizon Computer Services (504) 469-1105.

Effective Management Systems, Inc. Authorized Digital Equipment Distributor (414) 251-6662.

WANTED by Hobbyist, DEC PDP-11, LSI-11 Software, parts, boards, manuals, peripherals, etc. Working or not. H. Kolesnik, 5277 S. Kenton Way, Englewood, Colo 80111, (303) 779-5256.

ORACLE — complete relational DBMS — NOW AVAILABLE FOR RSTS! Call RSI (301) 657-4475.

Software Techniques, Inc., is happy to announce the hiring of two new employees — Wef Fleischman and Tom Mynar.

TIMESHARING AVAILABLE on PDP 11/70 RSTS. Software also. Call Wabash Time Share in Indiana (317) 448-1686.

FOR SALE (2) RK05F, (4) MS11-JP. Call Ron Maclean (414) 743-8282.

RSTS/E HEAVYWEIGHT

Needed as in-house expert for sysgens, updates, optimization; consultant for applications programmers. 20 + RSTS/E installations on various hardware configurations. Northeast location, some travel. Excellent salary and benefits. Gannett Co., Inc., Lincoln Tower, 15th FI., Rochester, NY 14604, (716) 546-8600.

DELAWARE VALLEY TIMESHARING Eastern Pennsylvania Philadelphia South Jersey **DEC 11/70 RSTS/E** WORD-11 24 hours - 7 days Most commerical packages Custom work. Raw time NATIONWIDE DATA DIALOG 70 James Way Southampton, Pa. 18966 215-364-2800 The FAMOUS

RSTS PROFESSIONAL TEE-SHIRT is now for sale!

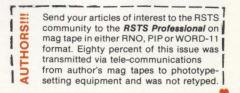
Send size desired and \$6.95 for each shirt to: RSTS TEE-SHIRT, P.O. Box 361, Ft. Washington, PA 19034-0361. Shirts available in adults sizes only: Small — Medium — Large — X-Large

Suburban Phila. financial institution wants experienced Basic + Programmer. Reply to TSO, c/o M Systems, Inc., Box 361, Ft. Washington, PA 19034.

Just how many [1,*] passwords do you use that are catchy and easy to remember?? * Lots of Luck from LOCK-11 (p. 49).

DEC VAX 780 Memory MS 780-DB [.2MB] available now very cheap (215) 728-2760.

SPREE — RSTS/E Extended Program Editor \$200 (503)644-8073.





Occasionally we are requested to print news that may be of interest to the RSTS community. We are happy to offer this feature to our readers. We reserve the right to print only as time and space permit. We cannot return photos or manuscripts. Send news releases to: RSTS News Release, P.O. Box 361, Ft. Washington, PA 19034-0361

July, 1981 SSI INTRODUCES FOUR NEW PRINTER SYSTEMS

Fort Lauderdale, FL — A new family of four line printer systems, The QT Series, has been introduced by Southern Systems, Inc., computer printer company.

The 300, 600, 1000 and 1200 lpm printer systems can be interfaced with computers from most manufacturers. Compatibility with the end-user's computer is guaranteed by Southern Systems

Parallel, synchronous serial or asynchronous serial operation of the printers is available.

The quietest standard line printer on the market, The QT Series is built to handle intensive data processing applications.

Operator-friendly standard features include operator-changeable bands, swing-open band gate, easy ribbon changing, excellent print quality, diagnostic display, membrane touch controls, form length select switch, 6/8 lpi and casters. The easy band change also automatically adjusts the printer to the new character set and to vertical alignment of the band. Special bands, including foreign language and OCR-A, are available.

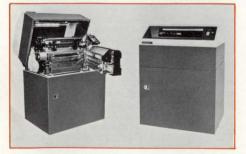
The QT series is easy to maintain and offers as standard such convenient servicing features as double opening bonnet — front and rear, double tilt paper bail, captured cabinet hardware, swing-out PC board rack, edge-connected PC boards, high ESD levels and internal self-test.

Standard configuration features also include a towel ribbon with deskew, casters, forms/alignment and column scale, paper motion verification, computer controller and I/O cable, single print control; and multi-band sensing.

The QT 300 and 600 lpm systems are available with both 132-column and 198column (condensed) formats.

The QT-3 prints at 220 lpm with a 96character set, at 300 lpm with a 64-character set and at 360 lpm with a 48-character set. The QT-6 prints at 440 lpm with a 96-character set at 600 lpm with a 64-character set; at 720 lpm with a 48-character set.

The QT-10 prints at 740 lpm with a 96character set, at 1000 lpm with a 64-character



Southern Systems new QT Series offers printer systems that operate at 300, 600, 1000 and 1200 lpm with computers from DEC, DG, HP, Burroughs, Texas Instruments and most others. Compatibility with the end-user's computer is guaranteed by Southern Systems. The quietest standard line printer on the market, The QT Series is built to handle intensive data processing applications.

set at 1200 lpm with a 48-character set. The QT-12 prints at 890 1pm with a 96-character set; at 1200 lpm with a 64-character set and at 1440 lpm with a 48-character set.

Southern Systems provides complete printer systems for end-users, systems houses and in the United States, United Kingdom and Europe. Speed ranges are from 200 lpm up to 1500 lpm.

Southern Systems headquarters are located at 2841 Cypress Creek Road, Fort Lauderdale, FL 33309. (305) 979-1000, (800) 327-5602. Telex 522135.

November, 1981

NEW uP WINCHESTER DISC CONTROLLER INTERFACES up to Two Seagate ST506 or ST512 Compatible Drives with LSI-11 and Offers DEC Software Compatibility Garden Grove, CA — A new low cost intelligent uP based Winchester Hard Disc Controller for interfacing two Seagate ST506 or ST512 compatibile drives with RL01/RL02 emulation, is now available from DILOG (Distributed Logic Corp.), for use with DEC LSI-11, 11/2 or 11/23 based computer systems.

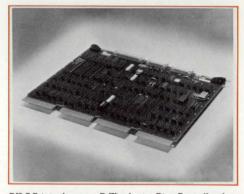
Designated the Model DQ604, the guad size board plugs into a single Q-bus slot for minimum space and power requirements.

In operation the controller runs under the RT-11 and RSX-11 operating systems using standard DEC RL01/RL02 drivers. A format diagnostic routine is supplied with each controller.

Other features of the DQ604 include an on-board multiple device hardware bootstrap * bad sector mapping when formatting * automatic flawed media compensation for hard errors on the disk * automatic read retry for soft errors * and full sector data buffer to eliminate DMA latency.

In addition, the controller includes an automatic self-test card edge LED to indicate successful microdiagnostics completion, and data protect to prevent erroneous data entry.

Price: \$2,050. each, Delivery: Stock to 45 days A.R.O., Direct inquiries to: Mr. Dennis Edwards, Nat'l Sales Mgr., DILOG (Distributed Logic Corp.), 12800 Garden Grove Blvd., Garden Grove, CA 92643, Phone: (714) 534-8950, Telex: 681 399 DILOG GGVE.



DILOG introduces a uP Winchester Disc Controller that interfaces two Seagate ST506 or ST512 compatible drives to DEC LSI-11, 11/2 or 11/23 based computer systems, and offers DEC compatibility.

September, 1981

ALTERNATIVE TO COSTLY CPU COMMU-NICATION NOW AVAILABLE FOR RSTS AND VMS

Bedford, MA — Clyde Digital Systems (CDS), a principal supplier of software for DEC systems since 1970, has recently made available for purchase a reasonably priced computer to computer communication utility named CALOUT.

Originally developed by CDS as a means for direct delivery of its own software products and updates to client companies, CALOUT has been recognized as a valuable tool for increasing the versatility for RSTS/E and VMS environments.

By making use of industry standard communication equipment, CALOUT provides a cost efficient means for moving all types of file between computers in either direction. At any baud rate, CALOUT's error checking capability insures that all transfers, including binary, reach their destination correctly without distortion

CALOUT's flexibility suggests that it provides an interesting alternative to more expensive file transfer packages. While built to facilitate inter RSTS or inter VMS dialogue, CALOUT's value is enhanced by its ability to allow non-binary data transfers to different operating systems and CPU's.

Clyde Digital Systems is now making CALOUT available for your inspection and evaluation. Sale on approval is being offered to qualified buyers for a 30 day price of \$25.00. The purchase price of this package is \$895.00.

For more information on CALOUT or other CDS products contact: Clyde Digital Systems, PO Box 348, Bedford, MA. (617) 275-6642.

October, 1981

SOFTWARE VENDORS NOW PERFORMING DYNAMIC, REMOTE, APPLICATION PACK-AGE DEMOS

Bedford, MA — Clyde Digital Systems, Inc., an international supplier of general application and utility software, reports widespread aquisition of their DEMO-11 utility package by software vendors. Released early this year, the product is currently in use at over 100 user sites with sales expanding rapidly.

According to the spokesman at Clyde Digital Systems, a vendor of an application package simply has the prospective customer dial up the vendor's computer as a normal remote user. The customer and vendor then make voice telephone contact. The customer evaluation team gathers around a terminal, remotely logged into the vendor's computer while they talk on the phone to the vendor. The vendor simply runs the DEMO-11 utility program and links his own terminal to the customer's remote keyboard. Now any task run by the vendor at his terminal will be seen at the remote, customer keyboard. It is as though the vendor were at the customer site in person, demonstrating the vendor's application products at the customer's remote keyboard.

Also, the customer may exercise the vendor application package at the remote site. This in turn will appear at the vendor's terminal exactly as it appears at the customer's remote keyboard. Both terminals respond identically to the task when either is exercised.

The spokesman at Clyde Digital Systems notes that any task whatever can be run on terminals linked with DEMO-11. This includes all the various keyboard modes and run time systems that may be invoked by the vendor's application package

The DEMO-11 product promises to offer considerable savings in the high cost of placing vendcr personnel at a customer site for the purpose of demonstrating an application package

DEMO-11 is currently implemented on the DEC PDP 11 series of computers under the RSTS operating system. Clyde Digital Systems offers an unusual warranty. It is unconditionally guaranteed to work perfectly. No maintenance contract is needed and no installation is required. The spokesman from Clyde Digital Systems further notes that new users of DEMO-11 become confident users in about 10 minutes

DEMO-11 may be delivered by dial up line to a new user's computer within 24 hours of receipt of order. Further information can be obtained from Bill Keefe at Clyde Digital Systems, Bedford, MA 07130. (617) 275-2924.

page 110

RSTSPROFESSIONALRSTSPROFESSIONA

August, 1981

VERSION 2.0 OF ROSS/V Lexington, MA — Version 2.0 of ROSS/V, a software package which provides a RSTS/E operating system environment on DEC's VAX-11 computers under VMS, is now available for use. Among the many new features of version 2.0 are:

(i) enhancements to ROSS/V's terminalhandling capabilities, including support for RSTS/E's binary input mode, "echo control" mode, multi-terminal I/O, and input and output escape sequence handling.

(ii) support for reading and writing DOS and ANSI file structured magtapes, as well as for specification of tape density and parity.

(iii) support for a number of RSTS/E "sys calls" not previously handled by ROSS/V, including assigning and deassigning of devices, zeroing a magtape, and setting terminal characteristics.

(iv) support for RSTS/E pseudo-keyboards. ROSS/V, which is written in VAX-11 MACRO, interfaces to programs running under it, in the VAX's PDP-11 compatibility mode, at the RSTS/E monitor call level. Therefore, it is capable of running RSTS/E user programs without regard to the language in which they are written. BASIC-PLUS, BASIC-PLUS-2, DIBOL, FORTRAN IV, and MACRO-11 are among the RSTS/E languages whih have been successfully used under ROSS/V.

ROSS/V version 2.0 is priced at \$10,000 for a single-CPU perpetual license, with a one-year warranty including software support and any update releases of ROSS/V during that period. ROSS/V is available from: (Eastern U.S.) Evans Griffiths & Hart, Inc., 55 Waltham St., Lexington, Mass. 02173, (617) 861-0670; (Central U.S.) Interactive Information Systems, Inc., 10 Knollcrest Dr., Cincinnati, Ohio 45237, (513) 761-0132; (Western U.S.) OnLine Data Processing, Inc., N. 637 Hamilton, Spokane, Wash. 99202, (509) 484-3400.

May, 1981

SSI INTRODUCES SYNCH/ASYNCH SERIAL COMMUNICATIONS INTERFACE

Fort Lauderdale, Fla. — Southern Systems, Inc. (SSI), FL-based computer printer firm, has announced development of a completely programmable Serial Communications Interface capable of either synchronous or asynchronous printer-computer operations.

The microprocessor-controlled interface was developed by engineers in the company's Clearwater manufacturing plant.

Southern Systems' new interface, termed the SI-9076, functions with the company's 200 lpm matrix printer, its B-Series printers ranging from 300 to 900 lpm and with the new line of QT band printers that the company is presently introducing.

Programmed for synchronous operation, the SI-9076 enables Southern Systems printers to function with IBM 2780/3780, Burroughs Poll and Select, Univac 1004, Honeywell VIP 7700 and other industry standard protocols.

"Since our newest interface is completely programmable, it can handle many protocols, as well as special purpose tasks," said James W. Rule, vice president/marketing. "In general, the serial communications interface allows one to implement a 'smart' printer on a serial communications network. In some cases the interface can replace communications processors."

The SI-9076 operates full or half duplex or as a null modem hookup. The synchronous serial data rate may be from DC to 64K baud. The asynchronous rate is from 50 baud to 19.2K baud. Modem control signals are provided. Provisions also are made allowing these signals to be "forced true" with switches, said Rule.

SSI's serial communications interface can perform standard or special code conversions (ASCII, EBCDIC, BAUDOT, IBM SELECTRIC and others). Sufficient programming area is provided so that a complete "Lookup Table" can be implemented and any character converted to any other character.

The interface is useful in systems that require data verification before printing. Special block checks, cyclical redundancy characters (CRC), or error-correcting techniques can be programmed. Parity checking is standard, but may be disabled.

The SI-9076 allows one to implement additional features in the printer that are not normally present. Features incoporated as standards within the 9076 include foldover, space compression, vertical TAB, Serial DAVFU and Wraparound.

In addition, the SI-9076 microprocessor can read the standard set up switches (USART Setup and Baud Rate). In the test mode, a printout of the board set-up is performed and other readable switches are provided for address, terminal number, etc. This allows any user to verify the exact configuration of his communications printer.

SI-9076 contains 5K bytes of RAM for data buffering, although all 5K does not have to be installed. This memory may be partitioned as a single buffer or multiple buffers. It is normally programmable with dual buffers so that a record is being printed while a record is received. By using Double Buffers, maximum printer speed is attained at all times.

Up to 2K bytes of RAM are is supported for programming the interface board. This is generally more than required to handle specific tasks and also provides code conversion area, "canned messages," etc.

A summary of fetures provided by the SI-9076: Z80A Microprocessor, 5K Bytes of RAM, 2K Bytes of EPROM, Double Buffering, Crystal Controlled Baud Rate Generator, Synch/Asynch USART, Concantinated Rs232 Input/Output for line sharing, Current Loop Input, Modem Control Signals, 32 Readable Switches, Printer Output Buffer.

Specifications summary: Z80A Microprocessor System; 2K Bytes of program area; 5K Bytes of buffer storage; Synch/Asynh USART; Synchronous baud rate - DC to 64K baud, the external clock must be supplied by the modem (or comm. system); Asynchronous baud rate switch selectable from 50 baud to 19.2K baud; Parity -- switch selectable for even, odd, or nor parity; stop bits - switch selectable for 1, 11/2, 01 2 stop bits; Self test/Reset switch - a SPDT momentary contact switch is provided to reset all electronics (A self test feature is incorporated within the 9076 board); Concantinated RS232 Input/Output - separate RS232 input and output connectors are provided (multiple terminals can share a common communications line); Current Loop Interface - a 20MA current loop interface is provided (60MA with resistor changes). Either the XMITER or receiver can be made active or passive (active provides a current source).

September, 1981

SATURN SYSTEMS RELEASES VERSION 4.0 OF WP SATURN

Minneapolis, MN — Saturn Systems, Inc. has just released version 4.0 of WP Saturn, a Word and List Processing package that runs on Digital Equipment Corporations PDP-11 and VAX series of computers. The enhancements include:

The enhancements include:

1. Formatting in the Editor to allow viewing of centering, tabs and the margins without leaving the editor.

2. A Master Menu to simplify document selection and editing.

3. Bold Face printing.

4. Footers.

5. In addition to automatic decimal numbering of paragraphs (i.e., 3.7, 3.7.1, ...), the system now has standard outline format as an option (i.e., Roman Numerals, Capital letters, Numbers, ...).

For more information contact Saturn Systems, Inc., 6875 Washington Ave. So., Suite 218, Minneapolis, MN 55435, (800) 328-6145 or (612) 944-2452.

December, 1981

PRINTER SHARING DEVICE Palo Alto, CA — The Model 9 printer sharing device from Ross Systems, Inc., is a free standing unit which allows one serial printer to be shared by up to nine computers, either locally or at a remote site. It polls each computer using X-on and X-off. It will work with any printer which sends X-off when its buffer is full and X-on when its buffer is empty. Additionally, it will detect printer offline through the data terminal ready signal.

The Model 9 carries a 180 day warranty and is available within 30 days after receipt of order.

Ross Systems, Inc., 1900 Embarcadero Road, Suite #208, Palo Alto, CA 94303, (415) 856-1100.



modern educator is now a reality. CMI is a computer program designed to provide computerized instruction, tests and remedial work to students. It adapts to each students ability allowing them to work at their own pace. Students log on and access the information using the EXPLORE function, an easy to use, interactive program. The educator can design multiple learning modules using the flexible CREATE function and monitor each student's performance via the REPORT function. CMI was designed by university faculty and is currently being used in secondary school, vocational school and universities.

To find out how CMI can benefit your institution contact: ELEX (The Effective Learning Exchange), P.O. Box 14, Medford, New Jersey 08055, (609) 654-1100.

October 1981

SOFTWARE PACKAGE TRANSFERS FILES BETWEEN TWO DEC MINICOMPUTERS Without Using Communications Interfaces London, England — Suitable for LSI-11's and VAX-11's as well as PDP-11's, including processors with different operating systems and different storage media.

A low-cost and easy-to-implement means of transferring files (data and/or programs) between two Digital Equipment Corp. PDP-11, LSI-11 or VAX-11 computers, is making its debut in the U.S. Called XOREN IPL-11, it is a

software package which enables file-transfers to be carried out asynchronously between any two machines from the three families, either on the same site or in different geographical locations.

Developed by Xoren Computing Ltd, of London, England, IPL-11 enables a two-way communications sytem to be set up for less than \$3,000 (excluding line costs) and put into operation as soon as the program has been installed on each machine.

Transfers can take place in both directions simultaneously at rates up to 9600 bits/second, over a direct line or telephone line; and can be initiated manually or, in certain circumstances, under the control of a user's applications program.

The package carries out and monitors the entire transfer process. It performs CRC errorcorrection to CCITT Recommendation V41 and, when it detects errors, re-transmits the block or blocks in which the errors were found.

A key feature of the package is that it eliminates the need to buy dedicated communications interfaces. Transfers are via a terminal I/O port on each computer. Thus, for interprocessor communications between two computers on the same site, each machine requiers only a standard DEC DL-11 or equivalent single-channel terminal interface card, or a single port on a mult-channel interface card such as a DZ-11 or DH-11.

For communications between compters on different sites, each processor's interface card requires only the addition of the necessary modem.

IPL-11 provides a number of operational advantages. It is media-independent, i.e., transfers data irrespective of the data storage devices in use on each system. it eliminates the need to transport disks and other media when transferring data on or off site, and it allows data to be guickly stored at a remote site for securty.

The package also allows multi-system access to expensive peripherals such as large disk systems; rapid updating of data and/or programs on computers at remote sites; communications between word-processing and information-processing systems, and instant distribution of informatio in electronic mail applications. OPERATING SYSTEM INDEPENDENT

Another attractive feature of IPL-11 is that it is largely operating-system independent. Versions are available for most major DEC operating systems, including RSX-11M, RT-11, VAX-VMS (RSX-11 compatibility mode), IAS and, most recently, RSTS/E. Each version can communicate with any other.

The version for RSX-11M supports indirect command files and has a Remote Activation facility which enables a user at one computer to transfer files to or from another without an operator being present at the other machine. An optional file-conversion utility, FLC-11, provides a one-pass conversion for applications where the two operating systems use different file formats

A further option, IPLLIB, enables transfers under the RSX-11M version, to be controlled by users applications programs. The package is supplied under a 5-year licence. A separate licence is required for each combination of cpu and operating system under which IPL-11 is to run. The two licenses required to link two operating systems cost \$1350 each whichever pair of operating systems is specified. For nlarger orders a system of discounts is applied.

Xoren is currently setting up a network of distributors in the US and Canada to market the product to OEM's and end-users.

Xoren Computing Ltd is an independent systems/software company formed in 1974. It has developed real-time computer systems and software for several large organisations in the UK, Europe and North America, including the British Post Office, I.T.T. and the British Columbia Telephone Company and has developed a number of commuications-oriented software

products.

For more information contact: Mr. John Jarvis, Xoren Computing Limited, 28 Maddox Street, London W1R 9PF, England. Telephone: LON-DON (01) 629 5932.

September, 1981

WHY SYSTEMS INTRODUCES DIGICALC Redmond, Washington — Our development staff, directed by Wayne Yarnall, has designed an electronic spreadsheet program to run on Digital Equipment Corporation computer systems

DIGICALC is perfect for doing budgets, financial modeling, pro forma financial statements, investment and resource management, and job costing. The strong success of Personal Software's VisiCalc shows a clear need for this type of program in commercial environments. DIG-ICALC meets this need and contains the enhancements that you would expect from an advanced program running on a more powerful computer

DIGICALC displays a tabular worksheet on the screen of a VT-100 terminal. The user writes equations, values, and textual data onto the worksheet by way of the keyboard and auxiliary keypad. The program calculates the equations entered and recalculates each time related data is added or changed. Worksheets can be saved, recalled, and consolidated, and reports can be printed as needed.

DIGICALC is designed to be very userfriendly with extensive HELP available on the terminal. An inexperienced operator can sit down at the terminal and learn to use DIG-ICALC in less than an hour.

As a DEC system user, you wil find DIG-ICALC useful at every executive work station in your organization. Give us a call for an on-site demonstration to evaluate DIGICALC for your business needs. WHY Systems Inc., 20915 NE 77th St., Redmond, Wash. 98052, phone: (206) 881-2331.

November, 1981

NEW RELEASE 3.0 "RABBIT-3", JOB ACCOUNTING AND PERFORMANCE MONI-TOR for DEC RSTS/E Users

West Palm Beach, FL - RAXCO Inc. announces the immediate availability of relase 3.0 of RABBIT-3, Job Accounting and Performance Monitoring Software System for Digital Equipment Corporation RSTS/E operating system users

The purpose of RABBIT-3 is to monitor system activities and create data that can be utilized by performance analysis (RABBIT-2) and/or billing (RABBIT-1) programs. RABBIT-3 output may also be utilized by Datatrieve, as well as Fortran, Cobol and Basic users.

RABBIT-3 is a stand-alone system written in PDP macro that runs in 6-7 K memory system degredation is minimal.

RABBIT-3 requires no sysgen and contains an auto-load parameter.

RABBIT-3 creates records for the following resource types:

 Job Records . . . contain systems resources utilized by job.

• CPU Statistics Record . . contain periodic statistics in percentages of CPU utilization.

• Disk Space Records . . . periodic records

containing the disk space available.

 Disk Catalog Record . . . all information by filename, for all files on system including public and private disks.

• Disk Statistics Records . . . contain disk utilization information for each disk.

RABBIT-3 basic price is \$2495 plus options. Rental is \$99 per month plus options.

RABBIT Systems are in world-wide use on VAX-VMS and PDP-11 RSTS computers. For more information contact: RAXCO, Inc., 3336 N. Flagler Dr., West Palm Beach, FL 33407. Phone: (305) 842-2115.

LIST OF ADVERTISERS

Aardvark Software Inc p. 23
Able Computer I.B. Cover
Amcor
American Used Computer
C.D. Smith & Associates, Inc
California Computer Group p. 106
Casher Associates Inc p. 89
Clyde Digital Systems p. 11
ComDesign, Inc. p. 25
Computer & Terminal Exchange
Data Processign Design, Inc pp. 68, 82, B. Cover
Datamedia, Corp
Datasouth Computer Corp
Dataware, Inc
Direct Inc
Distributed Logic Corp
EEC Systems
Effective Learning Exchange
Emulex Corp pp. 20-21
Enterprise Technology Corp
Enterprise Technology Corp
Finar
Hamilton Rentals
Hardcopy p. 55
Infinity Software Corp p. 9
Instor Corporation
Macro-Man
Manus Services Corp p. 77
McHugh, Freeman & Associates, Inc p. 92
Nassau Systems p. 83
National Peripherals, Inc p. 35
National Public Radio p. 104
Nationwide Data Dialog pp. 49, 90, 91
NCA Corp
North County Computer Services, Inc , pp. 47, 56-57, 85
On-Track Systems, Inc
Oregon Software
Plycom Services, Inc
Radgo Sales Company p. 83
Raxco Inc
Real Time Products
Ross Systems, Inc
Software Results, Inc
Software Techniques, Inc
Southern Systems, Inc
Spartin Systems
System Industries
T.F. Hudgins & Associates, Inc
Telecom Computer Systems, Inc
Theta Business Systems
Timeplex, Inc
Tridacor Systems, Inc
Unitronix Corp
WHY Systems, Inc p. 73
Xoren Computing Ltd p 45



DEC[®]COMPATIBLE PERIPHERAL CONTROLLERS

Dataram Corporation offers the industry's widest range of DEC-compatible peripheral controllers — from comparatively simple NRZI tape controllers to complex 300 MB storage module drive (SMD) controllers.

An impressive array of state-of-the-art controllers, all built around high-speed bipolar microprocessors. All software compatible with the host LSI-11[®], PDP[®]-11, or VAX[®] minicomputer...and all available now.

And Dataram's controllers are designed to save you money, and, more importantly, space — our controllers typically occupy half the space required for the comparable controller from DEC. Doing it with a level of performance that makes any member of this family worth looking at.

The chart shows our current family of peripheral controllers, growing every day. If you don't see the controller you need, we're probably working on it right now. Call us and discuss your requirements.



Princeton Road Cranbury, New Jersey 08512 Tel: 609-799-0071 TWX: 510-685-2542

CONTROLLER	DESCRIPTION	COMPATIBILITY		
C03	Cartridge disk controller	RK05		
C33	Cartridge disk controller	RK05		
T03	NRZI mag tape controller	TM11/TU10		
T04/N	NRZI mag tape controller	TM11/TU10		
T04/D	Dual density mag tape controller	TM11/TU10		
T34/N	NRZI mag tape controller	TM11/TU10		
T34/D	Dual density mag tape controller	TM11/TU10		
T36	Dual density mag tape controller	TM11/TU10		
S03/A	80MB/300MB SMD controller	RM02/RM05		
S03/A1	160MB SMD controller	RM02		
S03/B	80MB/300MB SMD controller	RK07		
S03/C	200MB/300MB SMD controller	RP06		
S03/D	96MB CMD controller	RK06		
S33/A	80 MB/300 MB SMD controller	RM02/RM05		
S33/A1	80 MB/160 MB SMD controller	RM02		
S33/B	80 MB/300 MB SMD controller	RK07		
S33/C	200 MB/300 MB SMD controller	R P06		
S33/D	96 MB CMD controller	RK06		
Products printed in red are LSI-11 Bus compatible. Products printed in black are UNIBUS® compatible for PDP-11 and/or VAX minicomputers.				

DEC, LSI-11, PDP, UNIBUS and VAX are registered trademarks of Digital Equipment Corporation.

Canada: Ahearn & Soper Ltd., 416-245-4848 • Denmark: Technitron ApS, 02 96 98 22 • Finland: Systek OY, (80) 73 72 33 • France: YREL, (03) 956 81 42 • Hungary/Poland/Rumania: Unitronex Corporation, WARSAW 39 6218 • Italy: ESE s.r.l., 02/607 3626 • Netherlands: Technitron b.v., (020) 45 87 55 • Sweden: M. Stenhardt AB, (08) 739 00 50 • Switzerland: ADCOMP AG, 01/730 48 48 • United Kingdom: Sintrom Ellinor Ltd., (0734) 65464 • West Germany: O.E.M.-Elektronik GmbH, 07 11-79 80 47 • Yugoslavia: Institut "Jozef Stefan", 263-261 • Australia/New Zealand: Anderson Digital Equipment, (03) 544-3444 • India: Infosystems Private Limited, 79281 • Israel: Minix Computers & Systems Ltd., 03-298783 • Japan: Matsushita Electric Trading Co., Ltd., 06 (282) 5111 • Talwan: Rabbit Associates, Ltd., 7219573-5 • Hong Kong: Automated Systems (HX) Ltd., 5-630256-9 • Malaysia: Automated Systems (M) Sdn Bhd., 773777 • Indonesia: P. T. Daya ASL, 584306 • Singapore: Automated Systems (PTE) Ltd., 2354133

ABLE VaxDZ clears up your data traffic jams.

You've been hurting over the problem with VAX interrupt capacity for some time. Now we have a DZ that breaks up the terminal handling bottleneck and keeps your VAX system in the fast lane. It's ABLE VAXDZ, the only DZ with an output buffer which lets you select any silo depth from 0 to 16 characters. With this novel feature, you can literally set the optimum performance level for your system.

That alone should clear up the traffic, but there's more! We've given ABLE VAXDZ an intelligent input silo two times as big as the standard DZ buffer. Both big and smart means doubling the VAX input data-handling capacity in some systems or providing dramatic improvement in every system all the way up to the maximum line configuration. We've even included a "data throttle" which allows any external device to control the clear-tosend (CTS) line and optimize its own data rate.

VAXDZ puts sixteen lines with modem control on a single hex-width board at one unit load and includes a panel which supports EIA only (an optional panel supports a mix of EIA/CL). Other features include an onboard LED display for pinpointing malfunctions automatically, an on-board self-test for immediate verification of system integrity and a variable PROM set for proprietary OEM applications.

Now, here is the best part. ABLE VAXDZ will match or beat DH performance in VAX systems without the addition of foreign software.

You don't have to be a hero to deserve an ABLE VAXDZ medal. Just be smart enough to use our new multiplexer. Write for details. We'll include information on the ABLE line of UNIBUScompatible products, as well as the MAGNUM[™] Series of computer systems.



UNIBUS

UNIBUS

the computer experts ABLE COMPUTER, 1732 Reynolds Avenue,

ABLE COMPUTER, 1732 Reynolds Avenue, Irvine, California 92714. (714) 979-7030. TWX 910-595-1729 ACT IRIN.

ABLE COMPUTER-UK, 74/76 Northbrook Street, Newbury, Berkshire, England RG13 1AE. (0635) 32125. TELEX 848507 HJULPHG.

PA

ABLE COMPUTER-GERMANY, Forsthausstrasse 1, 8013 Haar (Near Munich), West Germany. 089/463080, 463089.

VAX and UNIBUS are trademarks of Digital Equipment Corporation.

Responsive Word Processing. Take Our Word For It.





WORD-11 is a complete word processing system. It's responsive. It's powerful. And it's sharable on up to fifty terminals while running concurrently with data processing.

WORD-11 is talented, too. Designed to work on Digital's

family of mini-computers, WORD-11 has all the standard word processing functions. For more sophisticated requirements, WORD-11 provides multiple dictionaries for spelling error detection. Automatic index and table of contents creation. Text search and replace. User defined keys. User-controlled hyphenation. And automatic footnoting.

Included with comprehensive list processing, WORD-11 offers fast sorting. Flexible selection. And extensive math functions.

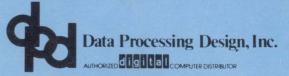
And WORD-11 has been proving itself for years. You'll find successful installations in small businesses, Fortune 500 companies, in universities and in banks-wherever Digital computers are in place.

Yet despite its sophistication, WORD-11 is easy to learn

and easy'to use. It's extremely cost effective. And it's available on RSX-11M, RSX-11M-PLUS, and RSTS/E. See how WORD-11 responds for you. Write for our brochure or call us direct.

• Corporate Office: 181 W. Orangethorpe Ave., Suite F, Placentia, CA 92670, (714) 993-4160, Telex 182-278.

• New York Office: 420 Lexington, Suite 647, New York, NY 10170, (212) 687-0104. · Washington D.C. Office: 4520 East-West Hwy., Suite 550, Bethesda, MD 20814, (301) 657-4098.



Overseas Distributors:

Management Information Services PTY. LTD. Melbourne, Australia

Jenson, LTD. Bristol, England

Sydney, Australia

Leeds, England

Network Computer Services PTY. LTD. Systime, LTD. On-Line Computing PTY. LTD. Subiaco, W. Australia