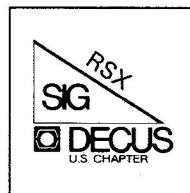
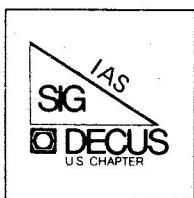




The DeVIAS Letter



The RSX Multi-Tasker

May 1984 Issue

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	PDT
DECnet	Digital Logo	RSTS
DECsystem-10	EduSystem	RSX
DECSYSTEM-20	IAS	UNIBUS
DECUS	MASSBUS	VAX
DECwriter	PDP	VMS
		VT

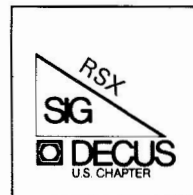
UNIX is a trademark of Bell Laboratories.

Copyright © Digital Equipment Corporation 1984
All Rights Reserved

It is assumed that all articles submitted to the editor of this newsletter are with the authors' permission to publish in any DECUS publication. The articles are the responsibility of the authors and, therefore, DECUS, Digital Equipment Corporation, and the editor assume no responsibility or liability for articles or information appearing in the document. The views herein expressed are those of the authors and do not necessarily express the views of DECUS or Digital Equipment Corporation.



Section One



The RSX Multi-Tasker

RSX MULTI-TASKER

Table of Contents

From the Editors	1
Where Did My File Go, PIP?	2
Statement of Menu Items	4
Working Group News	5
Library News	8
Help Yourself	9
RSX to IBM Floppy Transfer Needed	9
Jim Best (misc.)	9
It's On the SIG Tape	10
DTC	10
Hints and Things	11
Alternative to [1,2]LOGIN.TXT	
TDX Pool Depletion Problem	13
Three Routines to Ease Use of Printer	20
Placing a Quote in an Indirect Symbol	27

FROM THE EDITORS

This issue is a wee bit shorter than previous ones. We had a nice influx of things last month, but this month we didn't receive one article, and I have run out of things of my own to publish.

Two people who had promised me long articles failed to produce. I hope they will come through for next month.

Kitty Bethe of Citibank is doing an article tentatively titled "Happiness is Having a Warm PRO in Your Office" (no guys, not that kind of pro). She says that within the last six months the PRO/350 has become a real delight to use. The hardware works, software is available (things like Datatrieve, Runoff, TECO, and soon All-in-One from the VAX world), and -- undocumented -- all the hooks are there for making privileged RSX tasks run on the PRO. Should be an interesting article, and by printing this I have deliberately put her on the spot to produce.

We said in earlier issues that we would accept one or two page items in hard copy. That turns out to be more of a problem than we anticipated. One month we received about a dozen such items, totalling around twenty pages that we had to type in. Neither Dominic nor I are expert typists, so it took between four and five hours of our time. We try to do an entire issue in one day, and this really puts a crimp in our schedule.

Therefore we are going to have to make a hard and fast rule: nothing gets published unless it is sent to us in machine-readable format.

To speed things up, it's OK if you want to send it to me directly (address below) rather than through the DECUS office. And if you phone me I can give you a telephone number and current password to log in directly to my VAX and dump the file right into the Multitasker account. You must phone because we change the password frequently. If you have RSX Kermit from the last SIG tape, you can connect with Kermit on the VAX and have guaranteed error-free file transfer.

Allen Watson
Multi-Tasker Editor
THE RECORD
150 River Street
Hackensack, NJ 07602
(201)-646-4111

Where did My file Go, PIP?

For you RSX folks with nothing better to do, try this --

1. Type @TI: <CR>

Yes, indirect can be invoked interactively. You should get indirect prompt, AT.>

2. Type:

```
.OPEN TEST.FIL
.DATA ABCDEFG
.DATA HIJKLMN
```

3. Now, try this:

```
Type: PIP ti: = TEST.FIL
You should get a warning that PIP cannot access the file because it is already opened with write access.
```

4. Close the file and look at it with PIP

```
Type: .CLOSE
then PIP TI:=TEST.FIL
You should see the two lines of the file you created.
```

5. Reopen the file for append access and add more data to it.

```
Type: .OPENA TEST.FIL
.DATA 12345
.DATA 67890
```

6. Here's the clincher --

```
Without closing the file, use PIP ti issue a delete.
Type: PIP TEST.FIL;*/DE
```

7. Add some more data to the file --

```
(Remember, we never closed it.)
Type: .DATA ZYXWVU
.DATA 987654
```

8. Close the file and check to make sure it is there.

```
Type: .CLOSE
PIP TEST.FIL;*/LI
```

You probably got the response

```
PIP -- No such file(s)
```

9. Now exit indirect

Type: .EXIT

What happened was the file was marked for delete in step 6 but not actually deleted until the file is closed in step 8. That is why it was possible to add additional lines of data in step 7.

When I contacted the Customer Support Center/Colorado Springs to find out if they had heard anything about this problem, they proceeded to research the problem and submit an SPR on my behalf. The basic tone of that SPR was as follows:

SUGGESTION/ENHANCEMENT

MINOR SYSTEM IMPACT

Issue a warning to the person with the open file that it was just marked for delete.

I did have a nightmare about the warning message. It would read:

HEY YOU. GIVE UP. PIP just marked your file for delete

(Note the user friendly style of the lower case message.)

I did submit my own SPR regarding the matter when CSC/CS assured me that they could do nothing more. The tone of my SPR was as follows:

PROBLEM/ERROR

HEAVY SYSTEM IMPACT

Issue a warning to the user of PIP that the file was not marked for delete because it was opened with write access.

I have not yet received a response to my SPR (nor the one submitted on my behalf). However, I do hope the DEC personnel located in Maynard will view this situation with more sincerity than those in Colorado Springs.

Scott E. Gordon

Anchor Hocking
109 North Broad Street
Lancaster, Ohio 43130

Statement of Menu Items

Allen Jay Bennett
Clark Automated Systems
Battle Creek, Michigan

Editor's Note

The following list of items for the RSX menu was submitted by Allen Jay Bennett. I am not clear whether these are all the menu items or just the new ones; the list seems rather short to me. (AAW)

- o At one time, typeahead (3.2) was allowed on slaved or attached terminals. The decision was made to support only on attached. Some of our applications did not allow for attaching, but the terminals were slaved, and typeahead worked for us. Digital did publish a patch for 4.0.
- o Support of BREAK function of terminal interfaces (DL-11, DZ-11, etc.) for communication with foreign systems.
- o Add I- and D-space support under M-Plus to the FORTRAN/MACRO Symbolic Debugger.
- o Add a switch to DMO to dismount the disk even if checkpoint space is allocated.
- o QMG should allow set command to modify or display more terminal characteristics such as set speed.
- o QMG/SHR INIT or STOP waits if the device is attached by another task.
- o Add two switches to ACS:
 1. List everything checkpointed to a checkpoint file.
 2. Force something checkpointed to become not checkpointed.
- o Batch job flag page uses UIC of submitting account for flag page large letters.
- o Support for Digital DF03 modem is sorely lacking. Need ability to control automated facilities of modem.

- o Allow use of autobaud on non-remote lines.
- o Allow use of (Q-bus) RX02 without restriction on 22-bit systems.
- o Double-check with user before BAD and INI.
- o Add "catch-all" capabilities to IAS.
- o Modify QMG package and PRT... to allow record lengths in excess of 132. characters.

Working Group News

Jeff Hamilton
Working Group Coordinator
(214)457-4175

Date of this report: 11APR84

The working group chairmen are as follows:

RSX-11M Unsupported Versions:

Bill Burton
Texas Research Institute
1300 Moursand
Houston, Texas 77030

System Performance and Accounting

Paul Sorenson
AEP, Interactive Graphics
1 Riverside Plaza
Columbus, Ohio 43215

DECUS Library

Bruce Zielinski
RCA
Marne Highway M/S 138-2
Moorestown, N. J. 08057

SIG Tape Collection

Glen Everhart
RCA Government Systems Division
Route 38
Cherry Hill, New Jersey 08358

SRD

Bob Turkelson
NASA/Goddard Space Flight Center
Mail Code 935
Greenbelt, Maryland 20771

Data Acquisition, Simulation and Process Control (DASPC)

Allen J. Bennett
Clark Equipment Co.

P.O.Box 3000
Battle Creek, Mich. 49016
Runoff
Chuck Spalding
Adept Technology Inc.
1202 Charleston Rd.
Mountain View, Calif. 94043

The Unsupported Version working group is currently planning sessions for the Spring Symposium for past unsupported versions of llM/llM+. An article is being submitted to the Multitasker about the current work being done by the working group.

The System Performance and Accounting working group is continueing its work in preparing the index of the past RSXSIG tapes in regards to System performance and accounting. There is a session being planned for the Spring Symposium for the System Performance and Accounting working group to discuss further work to be done. Anyone interested in system performance and accounting please attend the working group session. Paul reports that as of April 30 his job responsibilities will be changing such that he will have less day-to-day contact with RSX. If there is someone who would be interested in taking over the chair of the working group please get in touch with Paul.

The DECUS Library working group have continued efforts to construct a tape to provide to the DECUS library of the best software off of the past RSXSIG tapes. A form letter was sent out to the 60 members of the working group and 30 replies were returned. These people are being used in the evaluation of a sample 2400' 800 bpi tape that will be submitted to the DECUS library. The RSX European branch will also be used to evaluate the software and will receive the tapes from Glenn Everhart (see below).

According to Glenn Everhart the SIG Tape working group has not done much. Copies of the tape were sent to DECUS UK and DECUS Germany to the RSX people there. Also tapes were being made available to those regional people from the National LUG organization interested in distributing the tape in parallel with the tree. This will be continued, and LUGS will be able to get the tape using either (or both) path. Since forms will go out with the tapes in either case to allow them to be identified (the forms should be returned to Glenn Everhart upon receipt of the tape so the Working Group can account for who has the tape), a LUG will be accounted for no matter how it gets the tapes. LUGS will vote by electing to be part of the tree or not on how big the tree will be. In the meanwhile we will not break the tree and will use the LUG organization in PARALLEL to it. (The VAX SIG is going to use this philosophy too, a change from earlier plans.)

The LUG organization has some highly reliable nodes and some less so. One of the regional coordinators' phones had been disconnected when I tried to call, and his LUG chairman complains

that he has rarely shown up at LUG meetings. Most of that group appears reliable though. The tape will be made available to any regional LUG librarian who sends Glenn Everhart a blank 2400' reel to ask for it, and this will continue.

Kitty Bethe, the Eastern Tree Coordinator, has many of the Sig tape packages running under P/OS. All the FCS stuff and most all privileged code for RSX runs unchanged under P/OS with simple relinking. Accordingly, a question has been placed in the form letter that goes out with the tree asking which LUGS have people interested in packaging SIG tape software onto RX50 locally. An article will appear in the Multitasker about this. Contributions are also being solicited to Tape Copy on RX50 or RX01/RX02 (or even TU58). We will only be able to copy tapes at the symposium, but other media will be returned later if submitted. They may be submitted BEFORE the symposium if possible too. This is the preferred course. Floppy contributions will be isolated somehow to allow their identities to be preserved for redistribution at LUGS from the tapes. The availability on the PRO 350 of the RSX SIG collection has made it important for PRO owners running P/OS to get involved in the RSX SIG; they will be able to learn tricks of getting software running on their PROs, and participate in one of the greatest software exchanges ever begun.

One of the other projects for the Working Group has been the cataloguing of the past SIG tapes. While Ralph Stamerjohn did this some time ago for the back tapes, nobody has updated that effort. He has offered to bring his index up to date, though, and we may within the next year or so get a new and updated list of the entire few hundred megs of software, cross indexed and KWIK'd.

The other major outstanding work is the release to the DECUS library of the pre-1979 tapes. These need to have any proprietary software on them cleaned off, and to have hopefully a little order imposed on their chaos. A merger of those tapes with redundant versions of files purged exists; it occupies about 250,000 blocks but is not well cleaned. Glenn is looking for someone who can do the work and who will consent to do it within some finite time period. Much of the older software is still worthwhile, and if nobody else volunteers, he may try to find time himself to clean the stuff up. Anyone who is willing to commit to getting the job done during calendar 1984 and who will work with Glenn on it should call at 609-338-6022 (that's Cherry Hill, N.J.) during work hours Eastern time.

Glenn also proposes "Hopefully we will be able to clear up our old agenda. I want to propose 2 new SIG agenda items: solicited software, and PRO involvement. The latter I have mentioned. Solicited software means I think we should suggest Useful Things that someone might write, offering at least public approbation for the contributor of them. One such idea is a network layer to underly the LBL Tools Mail that just uses modems to give us a free network. Another (due to Ralph Stamerjohn) is an automatic ODL

creator. Another is a windowing package for RSX (the terminal driver already has all needed hooks for an ACP-like thing to do this; anyone feel like writing one?). "

The SRD working group has continued work on the improved SRD supplied to the Fall 83 RSXSIG tape. Enhancements and continued bug fixes are being considered.

The DASPC working group has continued its efforts to lobby DEC for more real time development into RSX systems. This will continue into the future. Representatives from the LABS SIG, VAX Realtime Working Group and DASPC continued in their efforts to form a new SIG, but the status is unknown as of this writing. A separate session is also planned for this working group and should provide for an open forum in this area.

The Runoff group has continued its effort to consolidate desirable features of several versions of Runoff into an "official" version. Chuck, due to work efforts, is finding the time to upkeep the working group more and more difficult. If there is anyone willing to help in this position, please get in touch with Chuck. No sessions other than the general session is planned for the Spring Symposium.

If you are interested in providing information to a special working group concerning problems or ideas in that area, please get in touch with the working group chairman of that group.

Library News

Report on INDEX (DECUS Library No. 11-229)

INDEX 11-229 works great. There are some minor inconveniences with the open statement used in FORTRAN IV version 2-2.5 on RSX. It flags some items as an error if they appear as the last item on an initial line which is continued. DUPLEX 11-456 works but seems to get confused easily requiring restart. There is some sort of problem (not investigated) with sending files especially FORTRAN source to another computer. This may be because of improper RSX11M "set" commands to setup the remote port.

John C. Farmer
West Point Pepperell
Engineering Department
P.O. Box 71

West Point, Georgia 31833

Help Yourself

RSX to IBM Floppy Transfer Needed

Does anyone have a program like RT-11 FLX that will transfer from RSX to IBM "interchange" floppy formats? We now transfer RSX to RT with RSX FLX and RT to IBM with RT FLX. There must be a better way by now.

John C. Farmer
West Point Pepperell
Engineering Department
P.O. Box 71
West Point, Georgia 31833

Jim Best

Exxon Enterprises
US 276 & Old Laurens Road
Fountain Inn SC 29644

In February we converted from RSX-11M v4.1 to M+ v2.1. The sysgen went very smoothly and to date we have had no ill effects. We still have a problem logging Controller Detected State Parity Errors on RA80 (this has been on going for about a year and a half), so if anyone has any suggestions, please let me know. We have noticed that BRU seems to take less time to perform our daily and weekly backups as well as less tape.

I was finally able to build and load the Glenn Everhart 22-bit Memory Resident Disk Drive and am currently using this to hold the TOTAL data base logfiles. This has greatly improved the performance of TOTAL (I should have had them on another disk anyway because of head contention, but by using the memory drive I finally had an excuse to move them). I rebuilt several tasks, including

TOTAL, with FCSFSL and performance improved plus using less disk space. If there any RSX (Cincom)TOTAL users out there, I would like to hear from you to get and exchange ideas...

Also I am trying to write a (MACRO) routine that can be called from my programs to see if anyone not in my UIC is running my programs (i.e. if the logon UIC is not the same as mine, then the program will not execute). I would like some suggestions on how to do this. One might be, how to use \$TKTCB. I cannot seem to get it to work...

Having looked through the March Multi-Tasker, I am having the same problem of trying to order and receive materials from DEC Direct. Their service has been less than desirable.

One note to RSX Q & A number 9. by G. Abrahamian -- We had similar problems with F77 running RSX11M V4.0 on a PDP 11/44. We were able to over come the problem by reassigning the F77 work disks to a device other than our RA80. (That's were the memory disks is handy for us...)

*** P.S. ***

I just received from Ray Van Tassle the M+ version of FXDRV. I plan to use this in addition to Glenn's memory disk. My thanks to Ray for sending this...

It's on the SIG Tape

DTC

This utility is on the Fall 83 RSX SIG Tape and comes by way of Mitch Wyle and Glenn Everhart. It is written in FORTRAN. The following is a short document file that comes with it.

There is a new utility on the system called via the command DTC (for Desk Top Calendar). It works on VT100 terminals and will allow you to maintain your calendar in a much more visually obvious way than the old TOD and CAL commands do (though these remain useful).

To use it, you need an empty file in your account called DTC.DAT. The simplest way to create it is to use the command:

```
PIP DTC.DAT=NL:
```

When you enter DTC, you normally use commands D, W, or M for display of Day, Week, or Month. You give the value of the date as mmdyy (e.g. as 060283 for 6/2/83). (That's the only really

un-obvious thing about DTC). There's a built-in help command to tell more. Basically you can enter one liner schedule items for any half hour during the 8AM to 5PM interval and have the program flag them.

The month display flags days containing schedule items. The week display shows those plus the times occupied. The daily display gives the descriptions.

There is a Locate function to find times in a given week when a meeting could be scheduled, and the S command can schedule meetings for multiple users. To do this, a special file (normally other than DTC.DAT; the F command lets you change files used) with pointers to peoples' DTC.DAT files is used. The pointer entry is distinguished by having 99 in its' year field (see the furnished DTC.DAT for example) and the file specifier (terminated by an = sign) instead of an appointment. The S command will drop appointments into files so pointed to, but not into the current (pointer) file, while the D command drops appointments into the current file only, not pointers. the F command drops notices into both.

The default date/time starts out as today, but is whatever was last set up. There are 3 commands to give you a one day snapshot too:

```
DTC T {optional date}
    -- shows the daily display either for Today or for the
    date if one is included
DTC R {optional date in mmddyy form}
    -- shows the DTC weekly display (Reminder of scheduled
    meetings) for the current week or the one containing the
    day given if any
DTC C {optional mmyy}
    -- Shows the month currently or the one specified if one
    is given and flags busy days.
```

Hints and Things

Alternative to (1,2)LOGIN.TXT

Ralph W. Stamerjohn
Meridian Consulting
St. Louis, Missouri

In the LBL Software Tools is a feature that can be used as an alternative to placing system messages in the [1,2]LOGIN.TXT file. The LBL Software Tools mail supports a topic-specific mailbox. Such a mailbox, called a teleconference, is accessed by readers using the TC tool.

TC keeps track for each user of what messages they have read in each teleconference in a file in their account called tc.env. The TC tool has the following format:

```
tc [-<n>] [p|t] [topic] [entry]
```

The optional -<n> is the number of lines to display to a terminal before asking for a confirmation. The default is 22.

The p|t command specifies whether to print each message or list the message title. The default action is to print the messages.

The optional topic is the name of the teleconference. If not specified, the name 'general' is used. The keyword 'all' tells TC to use all conferences found in tc.env.

The optional entry is either a single message number, a message range in the form of 'n-m', or an open ended range in the form of 'n-'. The default action is to read from the message number saved in tc.env. When a new message is printed, tc.env is updated.

The simple command 'tc' prints all unread messages in the 'general' teleconference. The command 'tc t' lists the titles of all unread messages. By building a [1,2]SYSLOGIN.CMD file with the command 'shl -v tc t', all users will see the titles of unread messages in 'general'. They can then use the tc command to print messages they are interested in.

There are several advantages to the mechanism. The messages remain on file until the teleconference is purged. Therefore a user can always go back and reread an old message. If an account should never see the messages, create a dummy tc.env file and set a high last message number. Users can get a private copy of a message by redirecting output to a file, for example the following outputs messages 23-25 in 'general' to the file foo.bar:

```
tc p 23-25 >foo.bar
```

Messages are submitted to teleconferences using the Software Tools mail system. This leads to one of the most significant advantages of the system. The Software Tools mail system supports mail across networks, including DECnet between RSX and VMS systems. Further more an alias can be setup to allow a message to be routed

to many mailboxes (and thus teleconferences). Thus the address 'world' can mean the 'general' teleconference on an entire network of RSX and VMS systems. With a single operation, system managers can update the day message on their entire network.

Write and read access to a teleconference may be controlled. Therefore, while you allow everyone to read the day messages, only specific people can be given the ability to make a new entry if this is desired.

Teleconferences are extremely useful for other topics. I have used or see them used to keep track of software patch status, project logs, and hardware problem/downtime reports.

I have only encountered two possible disadvantages to the approach. First, there will be a pause during login while tc searches for unread messages. Outputting a status message like 'Checking for system status updates' seems to help.

The other disadvantage to this mechanism is the need to run the Software Tools mail system. While not needed by tc, the Tools mail would have to be running to make a new entry. If it does not run all the time, it is a simple matter to setup a command file to run the mail system, create messages, and turn mail off. My suspicion is that any system and network with enough users to benefit from this approach will also gain by using the tools mail for other communications also.

The LBL Software Tools are available on many SIG tapes, including the latest for both RSX and VMS. The Tools are in UIC's [307,30] to [307,36] on the RSX tapes.

TDX Pool Depletion Solution

Dr. Adrian Bottoms (RSX SIG Secretary)
Balanus Computing
28 Langdale Grove
Bingham, Notts NG13 8SR
United Kingdom

Editor's Note

This is part of a letter to Glen Everhart requesting assistance on getting the RSX SIG tapes distributed in the U.K.

As an advance gesture of returning some of the favour we owe I enclose some useful (??) items for publication in the Multitasker; one is the source of the TDX problem published in the January Software Dispatch. You will see that when I SPR'd it I also supplied a patch to fix the problem. The second is a patch for both the V4.0 and V4.1 full duplex terminal driver that prevents RSX hanging up if you have users who lean on the PF or cursor control keys and send MCR into a spin. The effect of the patch is to make TTDRV discard all UNSOLICITED input less than three characters long and terminated by an <ESC>. This works fine and does not affect anything.* TDX and CCL etc are OK since their command lines do not go through TTDRV* so even 1 or 2 character commands terminated by <ESC> work correctly. Note the patch is slightly different for V4.0 and V4.1.

Yours sincerely,

Dr. Adrian Bottoms (RSX SIG Secretary)

Fix to TTDRV (full-duplex) to stop pool depletion problems

A recurring problem for RSX users is pool (DSR) depletion resulting in a hung RSX system. Since the advent of pool monitoring tasks (first on the RSX SIG tapes and then officially from DEC with Version 4.0) these problems have at least been treatable if not avoidable (cure the symptoms if not the disease !!).

One source of pool depletion which has caused very rapid pool depletion in the past has been queueing of many command lines to MCR which results from holding down a PF key or cursor control key on a VT100 (or similar) terminal. This results in the VT100 generating a sequence of characters of the form <ESC>x or <ESC>xx. The first <ESC> is ignored by TTDRV as an empty command line leaving the xx hanging around. The next PF or arrow key terminates this with an <ESC> and results in the "line" being queued to MCR and also leaving another xx lying around. MCR generally does not understand xx<ESC> and tries to output a message to that effect (or ...CA. does eventually). This message is blocked from appearing if the terminal is not in full-duplex. The main result is that command lines queue up in pool waiting to be processed by MCR. MCR can not get rid of them rapidly enough and pool is quickly exhausted, since, on l1M systems, unsolicited input queued to MCR is sent in a 124 octal byte pool block.

One solution to this problem is to add support in ...CA. to
 * This statement is questionable. Some users (especially with KMSKIT) have patched MCR to allow such commands; they should be aware that this patch to TTDRV may lose that feature for them. It seems almost certain that some configuration of CCL with MCR and DCL might get into trouble with this patch. I cannot understand how Mr. Bottoms can say that CCL's commands do not use TTDRV.

discard command which it can recognize as originating from a keyboard escape sequence. This, however, adds a large overhead to the processing and is not very satisfactory. A much better solution is to discard commands of this form where they can first be recognized - in TDRV.

The patch given below provides support for this latter option. A patch is applied to the full-duplex terminal driver module [11,10]TTRW.MAC. The patch is placed so that all unsolicited input which is about to be queued to MCR is checked for two characters or less and terminated by <ESC>. If this test succeeds the input is discarded and a control-U is simulated at the users TI:. This patch adds 6 words of code executing in kernel mode. These six words save the allocation of an MCR buffer in pool (on 11M) - 124 octal bytes; and the considerable overhead of MCR processing an invalid command line.

NOTE

The patch comes in two forms: one for RSX11M V4.0 and one for RSX11M V4.1.

The only major restriction that this patch imposes on the system is that it is no longer possible to give meaningful 1 or 2 letter commands terminated by <ESC> to ...CA. VIA THE TERMINAL. Any other software should continue to work without problems since user applications, such as ...CA. and DEC applications such as ICP do not pass command lines through the terminal driver. It is still possible then to use 1 or 2 character commands terminated by <ESC> (to suppress the MCR prompt) without any problems. This patch only applies to UNSOLICITED input which would have gone to the user's CLI. All solicited input works exactly as before.

A SYSGEN is not required to install the patch.

1. Create the patch file in [11,60] as TTRW.SLP using the appropriate patch for V4.0 or V4.1.
2. Patch [11,10]TTRW.MAC with SLP.

```
SET /UIC=[11,10]
SLP @[11,60]TTRW.SLP
```

3. Re-assemble TTRW.

```
SET /UIC=[1,24]
MAC TTRW=LB:[1,1]EXEMC/ML,LB:[11,10]RSXMC,TTMAC,TTRW
```

4. Replace the TTRW module in TTDRV.OLB.

```
LBR TTRW.OLD=TTDRV/EX:TTRW
LBR TTDRV/RP=TTRW
```

5. Rebuild the terminal driver.

```
TKB @TTDRVBLD
```

6. VMR the new driver into the system.

```
VMR>UNL TT:
VMR>LOA TT:
```

7. Reboot the system.

```
BOO LB:[1,54]RSX11M
```

This patch has been in use here and at several other sites in the U.K. for several months on 11M systems without any problems or strange side effects.

This patch fixes the problems with TDX reported in Software Dispatch Jan-84. The patches were supplied to DEC with the original SPR. Why wait for update C? *** NOTE *** the MVZ instructions are not a typing error - it's a macro to move ASCIIZ strings around.

```
TDX.MAC;2/AU:80/-BF=TDX.MAC;1
\
-2,2
  .IDENT /12.AB/
-45
; Adrian Bottoms Balanus Computing 19-Jul-83
; AB001 a) Use MCR for SHQ command.
;       b) Prevent TDX crashing the system by terminating
;          the TPARS table with a blank STATE$ statement
;          as in the I/O Ops manual p7-14)
;       c) Make RAD40 "." work.
;       d) Allow less than 4 HEX digits in $HHHH.
;       e) Allow less than 3 RAD50 characters in %RRR.
;       f) allow "#" to be a metacharacter for <SPACE>
;          in RAD50 strings.
;       g) Update the ident message for TDX.
;
; Adrian Bottoms Balanus Computing 15-Oct-83
; AB002 Introduce the command ASY
;       ASY ==> ASN =SY:
;       ASY ddnn: ==> ASN ddnn:=SY;
;
%
-687,,/; AB002/
  COMAND ASY ; ASN ddnn:=SY: or ASN =SY:
-700,700,/; AB002/
TDXIDS: .ASCII <15><12>"TDX -- Version 12.AB (19-Jul-83)"<15><12>
-725,,/; AB002/
ASN: .ASCIIZ "ASN "
ASNSY: .ASCIIZ "=SY:"
-743,,/; AB002/
ASY: MOV #BIGBUF,R1 ; Point to spawn buffer.
     MOV #ASN,R0 ; Point to "ASN"
     MVZ R0,R1 ; Load command.
     TSTB 1(R5) ; ddnn: present ??
     BEQ 4$ ; No - Br.
     MVZ R5,R1 ; Yes - Load it.
     DEC R1 ; Strip trailing <CR>
4$: MOV #ASNSY,R0 ; Point to "=SY:"
     MVZ R0,R1 ; Load it.
     SUB #BIGBUF,R1 ; Get length of command.
     JMP SPNMCR ; Do it.

-773,773,/; AB001/
SHQM: .ASCIIZ "QUE /FU:ALL" ; Use MCR for SHQ command.
-1068,1073,/; AB001/
  TRAN$ !HEXSS,HEXSS2
```

```

TRAN$ $LAMDA,$EXIT,ACT6
STATE$ HEXSS2
TRAN$ !HEXSS,HEXSS3
TRAN$ $LAMDA,$EXIT,ACT6
STATE$ HEXSS3
TRAN$ !HEXSS,$EXIT,ACT6
TRAN$ $LAMDA,$EXIT,ACT6
-1102,1102,;/; AB001/
; Matches 1 to 3 RAD50 characters.
-1110,1119,;/; AB001/
TRAN$ !RADSS,RADS3
TRAN$ $LAMDA,$EXIT,ACT6
STATE$ RADS3
TRAN$ !RADSS,$EXIT,ACT6
STATE$ RADSS
TRAN$ '$,$EXIT,RAD3
TRAN$ '#,$EXIT,RAD5
TRAN$ '.,$EXIT,RAD1
TRAN$ $DIGIT,$EXIT,RAD1
TRAN$ $ALPHA,$EXIT,RAD2

STATE$ ; Terminate TPARS table !!!
-1405,1405,;/; AB001/
RAD5: CLR R2 ; Convert "#" to space.
BR RAD4 ; Join common code.
/

```

Three Routines to ease use of Printer

Raymond Willis

US Dept. of Agriculture
 Northeastern Forest Experiment Station
 Route 2 - Kentucky Highway
 21 East
 Berea, Kentucky 40403

There are three things that involve the printer that I have found useful and would like to share. The first involves spooling from within a program. The second is a subroutine to change the horizontal pitch and margin from within a program, and the third is a self contained program that allows one to give commands that make several changes on the printer somewhat similar to what the previously mentioned subroutine allows you to do except these

commands are given directly to MCR.

1. SPOOLING SUBROUTINE

DEC provides a Macro to spool output to a line printer but the Macro cannot be called directly from a Fortran program. This subroutine allows spooling from within a Fortran program. In order to use the subroutine, a file must be opened which will be written to during the course of the program. The statement for opening a file is: CALL ASSIGN(LUN,'NAME.DMP') Where LUN is the logical unit number to be written to for any text that is later to be spooled. NAME is the name of the file. Within the program each write statement should be the same as if you were writing to a terminal except the logical unit number as defined in the CALL ASSIGN statement is used. At the point in the program when spooling is desired, include the statement: CALL SPOOL (LUN,[IERR]) where IERR is an error number that you can look at or not depending on your mood. We use an LA120 as the printer to spool to. I assume that any printer will work. One MCR command that must be given before using this subroutine is: ASN TTN:=LP: where N is the terminal number of the printer.

```

        .TITLE   SPOOL SUBROUTINE FOR FORTRAN
;
;   FORTRAN CALL:
;       CALL SPOOL(LUN,IERR)
;
;       WHERE LUN = LOGICAL UNIT NUMBER
;
;   RETURN:
;       IERR = FCS ERROR CODE FROM FDB
;       IF IERR ARGUMENT IS NOT INCLUDED, STATUS CAN
;       BE PICKED UP AS FUNCTIONAL VALUE (FROM R0)
;   AUTHOR: Chris Heinz while employed by DEC under contract with
;           U. S. Forest Service, Berea, KY
;
;   NOTE:
;       FILE MUST BE OPEN
;
;   .MCALL   PRINT$                ; SEE CH. 8 OF I/O OP. MANUAL
;
;   .GLOBL   $FCHNL,$OTSV
;
;   .ENABL   LSB
SPOOL:: MOV     @2(R5),R2           ;GET LUN
        MOV     @#$OTSV,R3        ;GET WORK AREA ADDRESS
        JSR    PC,$FCHNL          ;FIND FDB ADDRESS
        MOV     R0,R1             ;SAVE ADR OF FORTRAN FDB
        ADD     #14,R0            ;GET TO RSX FDB
        PRINT$ R0                ;SPOOL FILE
        BCS    ERR               ;ERROR?
        MOV     #66,R0           ;THIS MANY WORDS TO ZERO
1$:     CLR     (R1)+            ;MAKE THIS LUN REUSABLE

```

```

                DEC      R0
                BNE     1$
2$:             CMPB    (R5),#1           ;ONLY ONE ARGUMENT?
                BEQ     3$               ;NO STATUS IN ERR ARG
                CMP     4(R5),#-1       ;ERR ARGUMENT DEFAULTED?
                BEQ     3$               ;NO STATUS IN ERR ARG
                MOV     R0,@4(R5)       ;STORE ERROR CODE OR 0
3$:             RTS     PC               ;AND RETURN TO CALLER
;
ERR:            MOVB    F.ERR(R0),R0    ;GET FCS ERR CODE FROM FDB
                BR     2$               ;AND USE COMMON RETURN
                .DSABL  LSB
                .END

```

2. SUBROUTINE TO CHANGE SIZE AND MARGIN

Another desirable feature is to be able to change the print size (horizontal pitch) and the margin from within a fortran program. My first attempt to do this resulted in a subroutine that worked fine as long as the output wasn't being spooled but didn't work any time the print out was being spooled. This subroutine, called CSIZE, changes both size and margin simultaneously and works irregardless of whether the print out is spooled or not. To use it have the line in the program:

```
CALL CSIZE(LUN,SIZE,MARGIN)
```

where LUN is the logical unit number, SIZE is an integer number from 1 through 8 where 1 is smallest print size (16 characters/inch) and 8 is the largest print size (5 characters/inch). The largest margin that this subroutine allows is a margin of 9. The subroutine can easily be modified, if a larger margin is desired.

Before the change in margin will take effect, there must be a carriage return output to the printer. Usually, people want the change to take effect immediately and so to accomplish this, the subroutine outputs that carriage return. The only time you would not want the change in margin to take effect immediately is when you want to change the print size in the middle of a line. In this case, modify the subroutine so that you do not output the CR at the end of the code. The comments in the subroutine show one way to modify it so the CR is not sent. With this modified version, the print size can be changed as often as desired within the same line and no change in margin occurs until you send the CR as part of your program.

```

C
C SETS THE CHARACTER SIZE AND MARGIN ON THE PRINTER
C LUN IS LOGICAL UNIT NO. ASSIGNED TO THE PRINTER
C IF ANY SIZE OTHER THAN 1 - 8 IS ENTERED, IT WILL SET SIZE TO
C SIZE 3 IF ANY MARGIN, OTHER THAN 1 - 9 IS ENTERED, IT WILL SET
C THE MARGIN TO 5 CAUTION: A CARRIAGE RETURN IS ISSUED AFTER THE
C MARGIN IS CHANGED. THIS IS NECESSARY IN ORDER THAT THE NEXT
C PRINTING HAS THE DESIRED MARGIN. AUTHOR: RAYMOND WILLIS,
C U. S. FOREST SERVICE, BEREA, KY
C
      SUBROUTINE CSIZE(LUN,SIZE,MARGIN)
      BYTE      NULL,PLUS,S1(4),SCODE(8),MAR(4)
      INTEGER   LUN,SIZE
      DATA     S1,SCODE/27,91,52,119,52,51,50,49,56,55,54,53/
      DATA     MAR,PLUS/27,91,53,115,43/
      IF(MARGIN.LE.0) MARGIN=5
      IF(MARGIN.GT.9) MARGIN=5
      MAR(3)=48+MARGIN
      IF(SIZE.LE.0) SIZE=3
      IF(SIZE.GT.8) SIZE=3
      S1(3)=SCODE(SIZE)
      IF (LUN .LT. 1) LUN=6
      IF (LUN .GT. 9) LUN=6
C      WRITE(LUN,100)NULL,S1,MAR      ! USE THIS TO CHANGE CHARACTER
C                                     ! SIZE IN MIDDLE OF LINE
      WRITE(LUN,100)PLUS,S1,MAR      ! USE THIS IF YOU DON'T WANT TO CHANGE
C                                     ! CHARACTER SIZE IN MIDDLE OF LINE
C
100  FORMAT(9A1)
C A WORD OF EXPLANATION IS PROBABLY IN ORDER HERE. THE FIRST
C CHARACTER IN THE ABOVE STRING IS THE PLUS SYMBOL WHICH CAUSES A
C CR TO BE OUTPUT AT THE END OF THE STRING. THIS IS NECESSARY
C BECAUSE A CHANGE IN THE MARGIN DOES NOT TAKE EFFECT UNTIL A CR
C HAS BEEN RECEIVED. IF YOU WERE TO SEND THE STRING WITHOUT THE
C CR AT THE END, YOU COULD CHANGE CHAR SIZE IN THE MIDDLE OF A
C LINE. THE FORTRAN MANUAL SHOWS HOW YOU CAN SEND A LINE WITH CR
C BUT NO LINE FEED, OR LINE FEED WITH NO CR BUT DOES NOT SAY HOW
C YOU CAN SEND A LINE WITH NEITHER CR NOR LINE FEED. THE I/O
C OPERATIONS MANUAL, SEC 2.8 (FOR RSX 3.2) POINTS OUT THAT A NULL
C CHARACTER TAKES AWAY BOTH CR AND LINE FEED. BY CHANGING THE
C VARIABLE PLUS TO THE VARIABLE NULL IN THE LINE ABOVE YOU WOULD
C BE ABLE TO CHANGE CHARACTER SIZE IN THE MIDDLE OF A LINE.
      RETURN
      END

```

3. COMMANDS THAT ALLOW PRINTER CHANGES TO BE MADE

The third useful submission is a program that allows any one of several commands to be given to the computer to allow changes to be made for the LA120. The following commands will work:

PRI LM n Set the left margin at position n on the printer.

PRI CS n Set the character size (horizontal pitch) to size n where n=1 is the smallest size (16 ch/in) and 8 is the largest (5 ch/in). If n is omitted or is any value other than 1 through 8, the default of 3 (12 ch/in) is selected.

PRI PA Advance to the next page. (Do a form feed.)

```

;                               PRI.MAC
;
; A PROGRAM TO SET CERTAIN PARAMETERS ON THE LA120 PRINTER. THE
; FOLLOWING COMMANDS CAN BE USED:
;
; PRI PA            ADVANCE TO NEXT PAGE
; PRI CS n         SET THE CHARACTER SIZE AT SIZE n. n = 1 IS THE
;                   SMALLEST. n = 8 IS THE LARGEST. IF n = 0 OR NO
;                   CHARACTER, SETS TO THE DEFAULT (12 CHAR/INCH, SAME
;                   AS 3)
; PRI LM n         SETS LEFT MARGIN AT VALUE n.
; Change the 5 in the ALUN$ macro to correspond to the terminal
; number of your LA120 printer.
; When task building, use the option
; TASK=...PRI
; After taskbuilding, install the task.
; Author: Raymond Willis, U. S. Forest Service, Berea, KY
;
;            .MCALL    GMCRS$,QIOW$$,EXIT$$,DIR$,QIOW$C,ALUN$$
GMCRC:    GMCRS$
OPT:      .ASCII    /PACSLM/            ; PG ADV, CHAR SIZE, LEFT MAR, R MAR
START:    ALUN$$    #4,#"TT,#5            ; LOG UNIT 4 IS THE PRINTER
          DIR$      #GMCRC            ; GET COMMAND LINE
          MOV       #GMCRC+6,R0       ; BEGINNING OF CHOICES
          MOV       R0,R1            ; SAVE IT IN R1
          MOV       #OPT,R2          ; GET BEGINNING OF OPTIONS
          CLR       R4
LET1:     CMPB      (R1)+,(R2)+       ; IS FIRST LETTER ENTERED = 1ST LET
;                   ; THIS CHOICE?
          BEQ       LET2            ; IT IS.
          INC       R2               ; SET FOR NEXT OPTION
          BR        NEXOPT
LET2:     CMPB      (R1)+,(R2)+       ; IS 2ND LETTER O.K.?
          BEQ       CHOI            ; IT IS, SUCCESS.
NEXOPT:   INC       R4               ; LOOK AT NEXT OPTION.
          CMP       R4,#3            ; LOOKED AT ALL?
          BEQ       ERR             ; WE HAVE
          MOV       R0,R1            ; GET BEGINNING OF BUFFER BACK
          BR        LET1
CHOI:     ADD       R4,R4            ; DOUBLE R4

```

```

        INC      R1                ; R1 NOW POINTS TO 1ST CHAR OF SIZE
        JMP      @TABLE(R4)        ; GO TO THE RIGHT CHOICE
ERR:    QIOW$$  #IO.WVB,#5,#1,,,,<#MES1,#LM1,#40> ; SYNTAX ERROR
        JMP      STOP
SIZER:  QIOW$C  IO.WVB,5,1,,,,<MES2,LM2,40> ; WRONG SIZE
        JMP      STOP
;
; CHOICE 1: SEND A FORMFEED
;
CHO1:   QIOW$C  IO.WVB,4,1,,,,<FF,1> ; PUT A FORMFEED TO PRINTER
        JMP      STOP
;
; CHOICE 2: SET THE CHARACTER SIZE
;
CHO2:   MOV     (R1),R1            ; SIZE SHOULD NOW BE IN R1
        CMP     R1,#60
        BLE     S3
        CMP     R1,#71
        BGE     SIZER
        SUB     #61,R1
        ADD     R1,R1            ; DOUBLE R1
        JMP     @SIZE(R1)
S1:    MOV     #64,SIZSTR+2
        BR     SIZGO
S2:    MOV     #63,SIZSTR+2
        BR     SIZGO
S3:    MOV     #62,SIZSTR+2
        BR     SIZGO
S4:    MOV     #60,SIZSTR+2
        BR     SIZGO
S5:    MOV     #70,SIZSTR+2
        BR     SIZGO
S6:    MOV     #67,SIZSTR+2
        BR     SIZGO
S7:    MOV     #66,SIZSTR+2
        BR     SIZGO
S8:    MOV     #65,SIZSTR+2
        BR     SIZGO
SIZGO: QIOW$C  IO.WVB,4,1,,,,<SIZSTR,4>
        BR     STOP
;
; CHOICE 3: SET THE LEFT MARGIN
;
CHO3:   MOV     #SIZSTR,R2
        ADD     #2,R2
C32:   MOV     (R1)+,(R2)+
        CMP     (R1),#60
        BGE     C32
        MOV     #163,(R2)+
        MOV     #15,(R2)+
        SUB     #SIZSTR,R2
        QIOW$$ #IO.WVB,#4,#1,,,,<#SIZSTR,R2>
;

```

```

;
STOP:   EXIT$$
TABLE:  .WORD   CHO1,CHO2,CHO3
SIZE:   .WORD   S1,S2,S3,S4,S5,S6,S7,S8
MES1:   .ASCII  /SYNTAX ERROR/
        LM1=.-MES1
MES2:   .ASCII  /THIS IS AN ILLEGAL CHAR SIZE/
        LM2=.-MES2
FF:     .BYTE   14                               ; THE FORMFEED
SIZSTR: .BYTE   33,133,60,167,15                 ; ESC, [, 0, w
        .END   START

```

The following example illustrates the use of the two subroutines. In this case the version of CSIZE that allows the print size to be changed in the middle of a line was used.

```

C
C          TEST.FTN
C
        BYTE   NULL
        DATA  NULL/0/
        CALL  ASSIGN(2,'JUNK.DMP;1')
        CALL  CSIZE(2,3,6)
        WRITE(2,100)
100     FORMAT(/,' THIS TEXT IS SIZE 3 WITH A MARGIN OF 6',/,
1' THE LINE THAT FOLLOWS SHOWS THE EFFECT OF REPLACING THE
2 PLUS CHARACTER',/, ' WITH THE NULL CHARACTER IN THE CSIZE
3 SUBROUTINE.')
        CALL  CSIZE(2,1,6)
        WRITE(2,110)           ! SEND A LINE FEED AT THE BEGINNING
110     FORMAT('$SIZE 1')      ! BUT NO CARRIAGE RETURN
        CALL  CSIZE(2,3,6)
        WRITE(2,120) NULL
120     FORMAT(A1,' SIZE 3')
        CALL  CSIZE(2,5,6)
        WRITE(2,130) NULL
130     FORMAT(A1,' SIZE 5')
        CALL  CSIZE(2,7,6)
        WRITE(2,140) NULL
140     FORMAT(A1,' SIZE 7')
        CALL  CSIZE(2,8,9)
        WRITE(2,150)           ! A CR IS SENT AT END OF THIS LINE
150     FORMAT('+ SIZE 8')    ! WHICH CAUSES NEXT LINE TO HAVE MARGIN 9
        CALL  CSIZE(2,3,9)
        WRITE(2,160)
160     FORMAT(' THIS IS ON A NEW LINE AND BACK TO SIZE
1 3 WITH MARGIN 9')
        CALL  SPOOL(2,IERR)
C  IERR WILL CONTAIN ANY ERROR CODE FROM THE SPOOLING
        CALL  EXIT
        END

```

Here is the output from the program.

```
THIS TEXT IS SIZE 3 WITH A MARGIN OF 6
THE LINE THAT FOLLOWS SHOWS THE EFFECT OF REPLACING THE PLUS CHARACTER
WITH THE NULL CHARACTER IN THE CSIZE SUBROUTINE.
SIZE 1 SIZE 3 SIZE 5 SIZE 7 SIZE 8
      THIS IS ON A NEW LINE AND BACK TO SIZE 3 WITH MARGIN 9
```

Placing a Quote in an Indirect Symbol

Roger Jenkins

Wycliffe Bible Translators
19891 Beach Blvd.
Huntington Beach, Ca. 92647

At a recent LUG meeting, while giving an introduction to Indirect talk, we got on to the subject of the difficulty placing a quote character (") in an indirect symbol. Ideally one would like to be able to type something like the following:

```
.SETS QUOTE """"
```

and have Indirect follow the commonly used rule that if a string delimiter occurs twice in succession within the delimited string, then one occurrence of the delimiter is to be stored as part of the string. Unfortunately, Indirect does not do this. (Developers, are you listening?)

It should be pointed out that this is really the simplest case of a larger problem. One would actually want to be able to place a quote anywhere in a quoted string, such as:

```
.SETS LINE "John said, ""My computer just broke."""
```

however once a single quote can be placed in a symbol, it could be appended to any other string in the appropriate place.

I suggested the following method of getting a quote character into a string symbol:

```
.ASKS QUOTE Please type a quote (") and press return
```

This has the disadvantage that you can't be guaranteed that the user will really type a quote and that it can't really be called "automatic". So I challenged others in the LUG to see what they could come up with.

A few days latter, Scott Snadow called me with the following suggestion:

```
.OPEN QUOTE.DAT  
.DATA "  
.CLOSE  
.OPENR QUOTE.DAT  
.READ QUOTE  
.CLOSE
```

A variation on his suggestion would be to have QUOTE.DAT always available in a known location (such as [1,1]) then the file wouldn't have to be created each time. The only potential problem I can think of would be if this was used in a subroutine and file #0 was already in use.

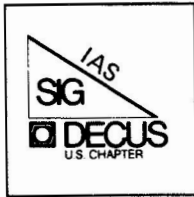
I think it would be very interesting to see how others have handled this problem. Perhaps the newsletter editor could sponsor an "Assign the Quote" contest with prizes going to the most unusual, simplest, most bug proof, and longest methods of assigning a quote to an indirect symbol.

From Editors

OK, Roger. Do you want to provide the prizes?



The DeVIAS Letter



Section Two

The DeVIAS Letter

The IAS SIG Newsletter

ISSUE No. 21

April - May 1984

In This Issue

Curley's Corner page 2

Letters

F. R. Borger (DPC Bug) page 3

K. J. Guralnik (JAP W.H.I.M.P.) page 11

J. Guidi (\$\$\$VRY) page 18

Articles

D. Pedlacek (A Generic PDP) page 22

R. Fussell (PRX) page 28

From the Editor

I can't help but feel that the end of an era has passed. The days of the PDP as the focus of large development projects is all but over. The days of DECUS as a group of concerned, informed users has ended leaving a beauracracy which, for the most part, has lost touch with large segments of its members.

The reasons were inevitable. Technology dictated better hardware and better software. Economics dictated the de-emphasis of small-growth product lines.

It is all very reasonable; but sad, nonetheless.

However, some of us still have PDP's and will continue to do our best to use the systems as effectively as possible. That is why this newsletter exists. Without your participation, in terms of submission of articles, ever this limited effort will fall by the wayside.

Please share your insight, your questions, your successes, your failures. Others can benefit as you have in the past.

Contributions

The DeVIAS Letter needs contributions in order to continue as an effective medium for exchange of information regarding IAS.

Contributions may be submitted in any form you wish. Originals on 8½ x 11 paper are preferred. However, even photocopies of relevant match-book covers would be appreciated.

Send all contributions to:

Ontario Hydro
700 University Avenue
Toronto, Ontario
CANADA, M5G 1X6

Attn: John W. Drummond
Mail Stop - M2E10

Department of Radiation Therapy
University of Pennsylvania
Room 410
133 South 36th Street
Philadelphia, Pennsylvania 19104
10 April 1984

Dear IAS Fan,

Do you ever get the feeling that IAS is an unknown operating system? I thought the phrase was 'little known', but I'm no longer sure. The guy from DEC Field Service was trying to find the flake problem with my 11/70 when he turned to me and said, "You'd be much better off running a DEC operating system." I got up on my soap box and let him have it, but it demonstrates something. Whatever...

A friend of mine is considering running RSTS/E on an 11/70 in a small college environment where almost all the users of the system are students. I argued that IAS has a better scheduler and would be more responsive to the terminal users than RSTS/E. No science, just prejudice. Can any of you assist my argument? Or, destroy it? He has an IAS license and a RSTS/E license - which should he use?

I had a phone call seeking FORTH on a PDP-11. Not necessarily an IAS version. Ever heard of any?

Also, the mass migration is continuing to VAX. There was a request for an operator's manual to help people who use IAS migrate to VMS. I haven't heard of one, have you? It sounds like a nice thing to share.

DECUS is gearing up for Cincinnati. Mike Reilly and I are giving a pre-symposium seminar on MACRO-11. We are both excited at the prospect, especially since there are already enough attendees to let it run. There are many of you who are experts (world class) in parts of IAS or other computer topics. How about presenting a seminar at Anaheim? How about a 1 hour session during the symposium?

Spring has finally come to Philadelphia. Maybe next time there will be more to say!

Happy Spring,
Robert F. Curley

March 30, 1984
Frank R. Borger
Instrumentation Division Head
Department of Medical Physics
Michael Reese Medical Center
29th Street and Ellis Avenue
Chicago, IL 60616

John W. Drummond
Ontario Hydro
700 University Avenue
Toronto, Ontario
CANADA, M5G 1X6
Mail Stop M2E10

Dear Mr Drummond:

We recently discovered a serious bug in DSC which we haven't seen reported, (although we dropped support some time ago, due to the parcity of reports and DEC'S failure to even answer our last SPR'S.) This may be of help to others out there.

Each night at midnight our system does automatic disk to disk DSC copies of our system disk or various user disks to a removable 67 mega-byte drive. Inspecting the printout one morning we were dismayed to see the following:

```
DSC - - *WARNING* 38 INPUT DEVICE DRO: FILE ID 006106,000112  
SEGMENT NUMBER CHECK
```

```
TASK "...DSC" TERMINATED  
T-bit trap or "BPT" execution  
PC=014566
```

DEC'S manual gave the explanation: The linkage connecting file segments was broken; the specified file is lost. Retry the operation.

The backup (of our system disk, no less,) had only proceeded about half way. We determined that the file in question was a temporary file that had not been properly deleted due to a system crash. We used PIP to reset the file,

```
PIP /FI:6106:71/DE
```

and waited for the next backup attempt. This time we got:

DSC - - *WARNING* 33 INPUT DEVICE DRO: FILE ID 006106,000112
IS DELETED

TASK "...DSC" TERMINATED
T-bit trap or "BPT" execution
PC=014566

.
DEC'S manual gave the explanation: The specified file was partially deleted on the input disk and was not copied. This is a warning only. No action is required.

Again, only about half the backup had completed. Using another trick we know, we re-used the file header by creating multiple copies of a dummy file with the editor until one of the versions used the offending file header. (We would do a TOF about 20 times, then exit the editor and do a full directory of the dummy files and check the file id numbers.) We then deleted all but the appropriate copy of the dummy file. We again sat back and waited. This time we got:

DSC - - *WARNING* 37 INPUT DEVICE DRO: FILE ID 006106,000112
SEQUENCE NUMBER CHECK

TASK "...DSC" TERMINATED
T-bit trap or "BPT" execution
PC=014566

.
DEC'S manual gave the explanation: The sequence number is incorrect. Retry the operation and/or replace the disk.

An examination of the file header using Glen Everhart's DFX showed it to be perfectly normal, there was nothing we could see wrong. Several attempts at changing it only produced either error 33 or error 37. Finally Dan had a stroke of genius and said, "Maybe it's a not a problem with file header 6106, maybe 6106 is an EXTENSION file header for some other file that's bad."

Searching through a 9420. block INDEXF.SYS file would be no easy task with DFX, that would take forever. I thought of doing a dump of the index file to another file and editing that to search for "6106", but a quick calculation showed the dump file would be about 5 times as big, 47000 blocks, so that was out. Thinking about what type of program could search the INDEX file, I realized I already had written a utility that almost did what I wanted. (Several DEVIAS members have copies of FHD, a program which will display the file header block for a file, given the FILE NUMBER and FILE SEQUENCE NUMBER.) A minor modification of this program produced FHX, a program which would sequentially search the IN-

DEX file for any file that had the header EXTENDED. FHX showed us that the file header for a 1-block command file had M.EFNU and M.EFSQ incorrectly pointing to file header 6106. (No way should a 1-block file have to use an EXTENSION FILE HEADER.) The extension file sequence number of course was not correct. We used DFX to clear the two offending location, re-calculated and reset the checksum, (we had modified Glen's DFX so that it could calculate checksums expressly for patching header blocks,) and wrote the file header back. Our next backup attempt FINALLY worked.

Had DSC reported the File Header Number for the bad header, rather than the one it incorrectly pointed to, we certainly would have found it a lot easier. In retrospective, that first error message rather vaguely pointed to a problem with an extension file header, but it certainly could have said it better. Oh-well, DSC'S diagnostic messages could be better, but anything is better than BRU.

For anyone interested, I have included a CMP output and a SLP command file so that anyone with a copy of our FHD program can create FHX easily.

Sincerely Yours,

Frank R. Borger

FRB/pdp

```

-1,6,;/FRB001/
      .TITLE  FHX
      .IDENT  /MRHVO1/
      .SBTTL  INTRO PAGE
;
;      THIS PROGRAM IS DESIGNED TO LIST OUT THE FILE HEADER BLOCK OF
;      ANY FILE THAT HAS EXTENSION HEADERS, TO AID IN FINDING PROBLEMS
;      THAT CAUSE DSC TO BOMB WITH "ID,SEQUENCE NUMBER CHECK" ERRORS
-11,14,;/FRB001/
;MCR>FHD
-19,,;/FRB001/
;
-35,46,;/FRB001/
-70,76,;/FRB001/
;      MOV      FID,BLOCKL      ;FIRST FILE ID IS BLOCK # IN INDEX FILE
      CLR      BLOCKL          ;START AT BLOCK 0
      ADD      OFFSET,BLOCKL    ;ADD OFFSET PREVIOUSLY FOUND
33$:  READ$    #FDB,,#BLOCKH    ;READ BLOCK N OF INDEX FILE
      BCS      ROPERR
      WAIT$    ;WAIT FOR READ
      BCS      ROPERR
      TST      READBF+136      ;EXTENSION FILE HEADER
      BEQ      34$            ;DONT PRINT IF NOT
      JSR      PC,PRINT        ;IF OK, PRINT FILE HEADER BLOCK
34$:  INC      BLOCKL          ;GET NEXT BLOCK
      BR       33$
-83,,;/FRB001/
;      ;JUST END OF FILE
-88,88,;/FRB001/
      MOV      #READOV,ERRQIO+Q.IOPL
-150,207,;/FRB001/
      RTS      PC
-227,228,;/FRB001/
READOV: .ASCII  /** FHX END OF INDEX FILE **/
      READLN=-.READOV
-291,296,;/FRB001/
      .ASCIZ  / /
/

```



```

*****
1) FHD.MAC;111
 1 .TITLE FHD
 2 .IDENT /MRHV01/
 3 .SBTTL INTRO PAGE
 4 ;
 5 ; THIS PROGRAM IS DESIGNED TO LIST OUT THE FILE HEADER BLOCK OF A
 6 ; GIVEN FILE
 7 ;

```

```

*****
2) FHX.MAC;7
 1 .TITLE FHX
 2 .IDENT /MRHV01/
 3 .SBTTL INTRO PAGE
 4 ;
 5 ; THIS PROGRAM IS DESIGNED TO LIST OUT THE FILE HEADER BLOCK OF
 6 ; ANY FILE THAT HAS EXTENSION HEADERS, TO AID IN FINDING PROBLEMS
 7 ; THAT CAUSE DSC TO BOMB WITH "ID,SEQUENCE NUMBER CHECK" ERRORS
 8 ;

```

```

*****
1) FHD.MAC;111
 11 ;MCR>FHD NNNNNN,MMMMMM WHERE
 12 ;
 13 ; NNNNNN,MMMMMM IS THE FILES (UNIQUE) FILE ID (AVAILABLE BY A FULL
 14 ; PIP OR SRD LISTING
 15 ;

```

```

*****
2) FHX.MAC;7
 12 ;MCR>FHD
 13 ;

```

```

*****
1) FHD.MAC;111
 20 .MCALL QIOW$,DIR$,EXIT$$,GMCR$,FHDOF$

```

```

*****
2) FHX.MAC;7
 18 ;
 19 .MCALL QIOW$,DIR$,EXIT$$,GMCR$,FHDOF$

```

```

*****
1) FHD.MAC;111
 35 MOV #GETMCR+G.MCRB+4,RO ;BUMP POINTER TO COMMAND PART
 36 JSR PC,$COTB ;CONVERT FIRST FID NUMBER
 37 CMP R2,#', ;TERMINATED ON "," ?
 38 BNE 222$ ;BR IF NOT
 39 MOV R1,FID ;SAVE FIRST PART OF FILE ID #
 40 JSR PC,$COTB ;NOW CONVERT 2ND PART OF FID
 41 CMP R2,#15 ;TERMINATED ON A CARRIAGE RETURN
 42 BEQ 1$ ;IT'S OK
 43 CMP R2,#33 ;OR AN ESCAPE
 44 BEQ 1$ ;THAT'S OK TOO
 45 222$: JMP SYNTER ;ANYTHING ELSE IS AN ERROR
 46 1$: MOV R1,FID+2 ;SAVE 2ND FID #
 47 FIRST:

```

```

*****
2) FHX.MAC;7
 34 FIRST:

```

```

*****
1) FHD.MAC;111
70      MOV      FID,BLOCKL      ;FIRST FILE ID IS BLOCK # IN INDEX FILE
71      ADD      OFFSET,BLOCKL   ;ADD OFFSET PREVIOUSLY FOUND
72      READ$    #FDB,,,#BLOCKH  ;READ BLOCK N OF INDEX FILE
73      BCS      ROPERR
74      WAIT$    ;WAIT FOR READ
75      BCS      ROPERR
76      JMP      PRINT           ;IF OK, PRINT FILE HEADER BLOCK
77      ;
*****
2) FHX.MAC;7
57      ;      MOV      FID,BLOCKL ;FIRST FILE ID IS BLOCK # IN INDEX FILE
58      CLR      BLOCKL          ;START AT BLOCK 0
59      ADD      OFFSET,BLOCKL   ;ADD OFFSET PREVIOUSLY FOUND
60      33$:    READ$    #FDB,,,#BLOCKH ;READ BLOCK N OF INDEX FILE
61      BCS      ROPERR
62      WAIT$    ;WAIT FOR READ
63      BCS      ROPERR
64      TST      READBF+136      ;EXTENSION FILE HEADER
65      BEQ      34$            ;DONT PRINT IF NOT
66      JSR      PC,PRINT        ;IF OK, PRINT FILE HEADER BLOCK
67      34$:    INC      BLOCKL   ;GET NEXT BLOCK
68      BR      33$
69      ;
*****
1) FHD.MAC;111
84      CLOSE$  #FDB           ;CLOSE IT OUT
*****
2) FHX.MAC;7
76      ;      ;JUST END OF FILE
77      CLOSE$  #FDB           ;CLOSE IT OUT
*****
1) FHD.MAC;111
88      MOV      #READFA,ERRQIO+Q.IOPL
89      MOV      #READLN,ERRQIO+Q.IOPL+2
*****
2) FHX.MAC;7
81      MOV      #READOV,ERRQIO+Q.IOPL
82      MOV      #READLN,ERRQIO+Q.IOPL+2

```

```
1) FHD.MAC;111
150 .PAGE
151 .SBTTL PRINT RETRIEVAL POINTERS
152 RETR: ;NOW GET RETRIEVAL POINTERS
153 MOV #READBF,R5 ;POINT TO BUFFER AGAIN
154 MOVH H.MPOF(R5),R4 ;WORD OFFSET TO MAP AREA
155 ASL R4 ;MAKE IT BYTE OFFSET
156 ADD R5,R4 ;NOW POINTS AT MAP AREA
157 MOV M.EFNU(R4),EFID ;SAVE EXTENSION FILE ID
158 MOV M.EFSQ(R4),EFID+2 ;AND SEQUENCE NUMBER
159 MOVH M.USE(R4),R5 ;LOOP COUNT FOR RETRIEVAL POINTERS
160 BIC #177400,R5 ;CLEAR POSSIBLE SIGN EXTENSION
161 ASR R5 ;NOW HAVE # OF POINTERS
162 BEQ DONE ;SKIP THIS IF 0 LENGTH FILE
163 ADD #M.RTRV,R4 ;R4 POINTS AT FIRST POINTER
164 22$: MOV #ARGBLK,R0 ;POINT TO ARG BLOCK
165 MOVH (R4)+,R2 ;GET HIGH BLOCK NUMBER
166 MOVH (R4)+,R3 ;GET SIZE
167 BIC #177400,R3 ;CLEAR POSSIBLE SIGN EXTEND
168 INC R3 ;CORRECT SIZE
169 MOV R3,(R0)+ ;PUT IN SIZE
170 MOV #DOTTAB,(R0)+ ;ADDRESS OF "<HT>" STRING
171 MOV R2,(R0)+ ;PUT HIGH BLOCK # IN ARG BLOCK
172 DEC R3 ;CHANGE SIZE TO OFFSET TO LAST BLOCK
173 ADD (R4),R3 ;MAKE R3 LAST BLOCK #
174 ADC R2 ;ADD CARRY TO HIGH BLOCK NUMBER JUST IN CASE
175 MOV (R4)+,(R0)+ ;PUT IN LOW BLOCK NUMBER
176 MOV #TAB,(R0)+ ;PUT IN "<HT>" ONLY
177 MOV R2,(R0)+ ;PUT IN LAST BLOCK HIGH BLOCK NUMBER
178 MOV R3,(R0)+ ;AND LAST BLOCK LOW BLOCK NUMBER
179 MOV #ARGBLK,R2 ;POINT BACK AGAIN
180 MOV #MESS3,R1 ;MESSAGE AREA
181 MOV #OUTBLK,R0
182 JSR PC,$EDMSG
183 MOV R1,ERRQIO+Q.IOPL+2
184 DIR$ #ERRQIO ;WRITE A RETRIEVAL POINTER
185 SOB R5,22$ ;AND DO N OF THEM
186 TST EFID ;WAS THERE AN EXTENSION FILE ?
187 BEQ DONE ;NO
188 MOV EFID,BLOCKL ;CHANGE TO EXTENSION FILE BLOCK NUMBER
189 ADD OFFSET,BLOCKL ;ADD OFFSET PREVIOUSLY FOUND
190 READ$ #FDB,,,#BLOCKH ;READ BLOCK N OF INDEX FILE
191 BCC 12$ ;OK
192 JMP ROPERR
193 12$: WAIT$ ;WAIT FOR READ
194 BCC 13$ ;OK
195 JMP ROPERR
196 13$: MOV #OUTBLK,R0 ;MESSAGE THE OUTPUT LINE
197 MOV #MESS4,R1
198 MOV #EFID,R2
199 JSR PC,$EDMSG
200 MOV R1,ERRQIO+Q.IOPL+2 ;UPDATE LENGTH
201 DIR$ #ERRQIO ;REPORT EXTENSION FILE HEADER
202 JMP RETR ;AND DO MORE
203 DONE: MOV #FINMES,ERRQIO+Q.IOPL
204 MOV #FINLEN,ERRQIO+Q.IOPL+2
205 DIR$ #ERRQIO ;WRITE FINAL LF MESSAGE
206 CLOSE$ #FDB ;CLOSE INDEX FILE
207 JMP EXIT ;AND WE'RE DONE
208 .PAGE
*****
```

```
2) FHX.MAC;7
143 RTS PC
144 .PAGE
```

```

*****
1) FHD.MAC;111
227 READFA: .ASCII /** FHD FAILED TO FIND FILE HEADER ***/
228 READLN=-.READFA
229 .EVEN
*****
2) FHX.MAC;7
163 READOV: .ASCII /** FHX END OF INDEX FILE ***/
164 READLN=-.READOV
165 .EVEN
*****
1) FHD.MAC;111
291 .ASCII /LIST OF RETRIEVAL POINTERS%N/
292 .ASCIZ /%NBLOCKS FIRST LAST%N/
293 .EVEN
294 MESS3: .ASCIZ /%M%2A%M,%P%A%M,%P/ ;SIZE,".<HT>" FIRST BL <HT> LAST BL
295 .EVEN
296 MESS4: .ASCIZ /%NFILE HEADER EXTENDED IN (%P,%P)%N/
297 .EVEN
*****
2) FHX.MAC;7
227 .ASCIZ / /
228 .EVEN

227 .ASCIZ / /
/ %N
FHDFHX.DIF=FHD.MAC,FHX.MAC

```

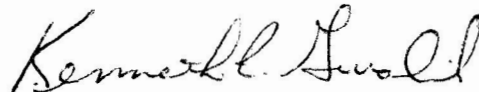
4 April 1984

Mr. John Drummond
Ontario Hydro
700 University Avenue (M4E5)
Toronto, Ontario M5G 1X6

Dear John:

Enclosed are two articles for the DEVIAS newsletter. The first is a copy of the WHIMs list sent to DEC. The second is another request for WHIMs. I have finally figured out how to get something to you. I can be contacted at (702) 295-2921 if you have any problems.

Very truly yours,



Kenneth I. Guralnik
EG&G Energy Measurements, Inc.

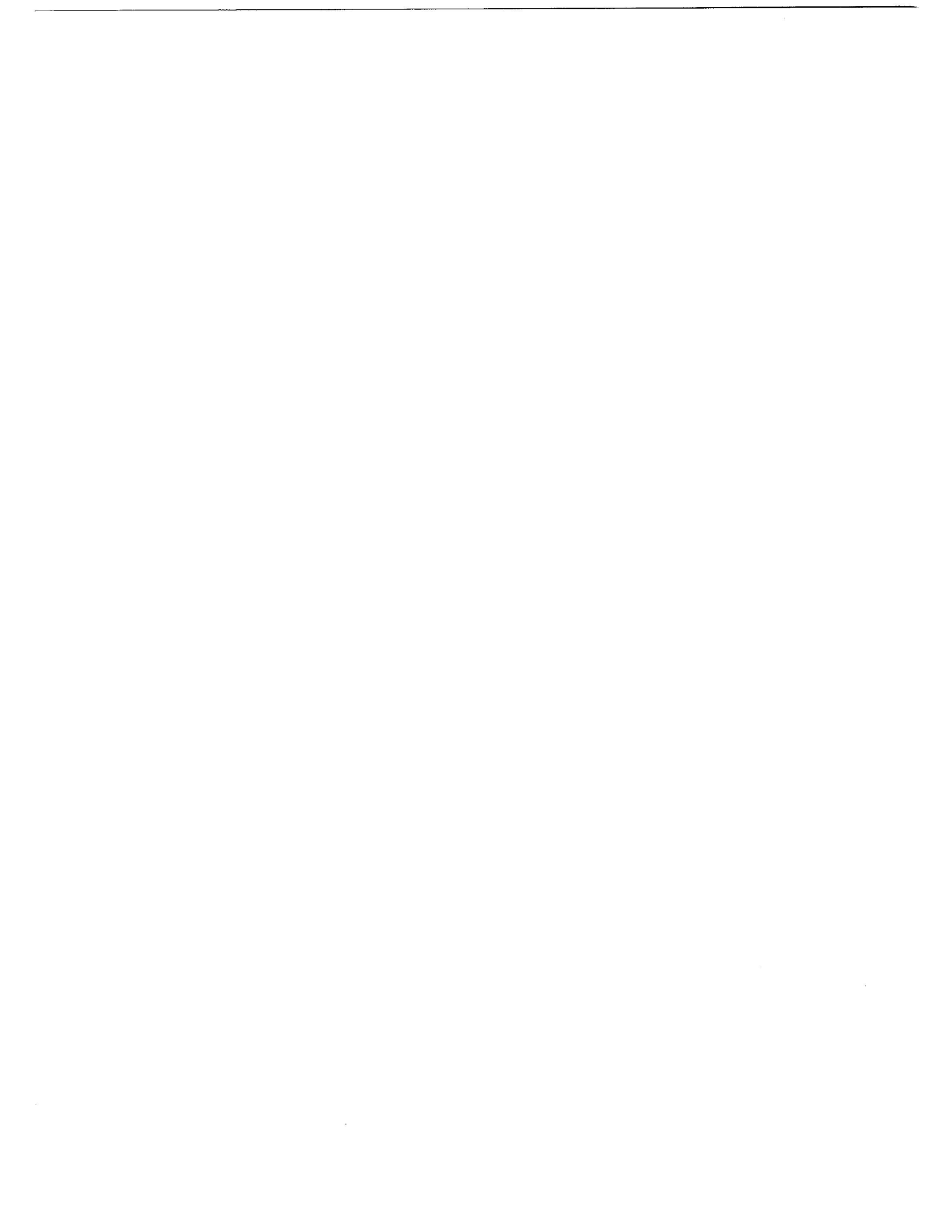
KIG:jg

To: IAS User's
From: WHIMs Commissioner
Subject: IAS WHIMs

We need your help! If any of you have a Wish, Hope, Idea, Muse or suggestion for IAS, please fill out the form (or the back of a cocktail napkin if necessary) and send it to me. I'll take anything; from the sublime to the ridiculous. We'll let DEC decide what is not feasible.

Also, any suggestions for DECUS sessions that you would like to be presented by DEC would also be greatly accepted.

Please help us!



IAS WHIMS

WISHES HOPES IDEAS MUSES SUGGESTIONS

IMPACT

IAS VERSION: _____ IAS FLAVOR: ALL R/T M/U T/S

CLI: PDS/PDX MCR CPU: _____

System Service _____ Utility _____

Layered Product _____ Other _____

WHIM

Name _____

Company _____

Address _____

Phone () _____

WHIMS Commissioner:

Ken Guralnik

EG&G Energy Measurements, Inc.

P.O. Box 1912, M/S S-05

Las Vegas, NV 89125



IAS WHIMs

WHIM: Upgrade to DECnet Phase IV.

Name: George Wells, General Electric Co., 502-452-5061

WHIM: Allow system manager to select at least some of the tasks that are controlled by the privilege bits.

Name: Ken Pauwels, Psychometric Lab UNC, 919-962-5085

WHIM: Support separate I&D space to allow for more program space and reduce overlaying.

Name: Steve Sizemore, UC San Francisco, 415-666-1344
Robert Mack, Informatics General Corp, 402-291-8300

WHIM: Provide a "Checkpoint Common Region" directive to save critical data maintained in common to facilitate restart.

Name: Robert Mack, Informatics General Corp, 402-291-8300

WHIM: Could the shuffler be made available on a real-time system only? Currently we gen as M/U bit disable the scheduler.

Name: Robert Mack, Informatics General Corp, 402-291-8300

WHIM: Extend the QIO directive to include timeout specification on all devices not just the TT handler?

Name: Robert Mack, Informatics General Corp, 402-291-8300

WHIM: Include EDT V3.0 in IAS.

Name: David VanderBrooke, Nabisco Brands Inc., 203-762-2500

WHIM: I do not like IAS to boot logged in. It is a huge weakness in the security of my system.

Name: Robert F. Curley, University of Pennsylvania, 215-662-3083

WHIM: Specify requested exit AST. Declare an AST when a task is being aborted to allow the task to exit gracefully.

Name: Tom O'Keefe, UTC Mostek, 214-466-7289

WHIM: Publish "hooks" into PDS for pre and post-session processing.

Name: Robert Osbrink, International Harvester, 312-932-5389

WHIM: Provide an indirect command file processor like IND to process PDS commands instead of MCR.

Name: Robert Osbrink, International Harvester, 312-932-5389

WHIM: Allow for reading of files in IND.

Name: Ken Guralnik, EG&G E/M, 702-295-2921

WHIM: Provide a switch to MOU to suppress the printout of the device information.

Name: Ken Guralnik, EG&G E/M, 702-295-2921

WHIM: Provide a switch of PIP to perform selected deletes when purging (/PU/SD).

Name: Ken Guralnik, EG&G E/M, 702-295-2921

WHIM: Enable the removal of installed tasks that have been deleted and the headers over-written.

Name: Kathy Good, EG&G E/M, 702-295-2940

WHIM: Provide an option to QUE to display currently printing files and remain active and print each new printing file until no files are queued.

Name: Robert Jones, EG&G E/M, 702-295-2941

WHIM: Better documentation on overlays and reading maps.

Name: Tami Lankford, EG&G E/M, 702-295-2939

WHIM: Allow for the writing/reading of VIRTUAL ARRAYS directly in QIO's from Fortran '77.

Name: Ken Guralnik, EG&G E/M, 702-295-2921

IAS SESSION WHIMs

SESSION WHIM: "Perhaps the best of the new directives added to RSX" (Hows and whys of AST's in RSX, 3.2)

Name: Tom O'Keefe, UTC Mostek, 214-466-7289

SESSION WHIM: Use of TCS macros and required user privilege.

Name: Robert Osbrink, International Harvester, 312-932-5389

SESSION WHIM: Using supervisor mode.

Name: Robert Osbrink, International Harvester, 312-932-5389

John Guidi
The Computing Center
The Jackson Laboratory
Bar Harbor, Maine
04609
phone: (207)288-3371 X-391

Bob;

We are having an interesting experience I would like to share with you. Something is eating our file system! Files which are accessible one day suddenly develop file-id, sequence number errors the next. Literally hundreds of files were being affected.

Examination of ERRLOG showed no reported hardware problems. VFY was useful to the point that it confirmed we had some problem, but gave little information to help isolate it. I looked through the Software Dispatches, the DECUS tapes, the MULTI-TASKERS, and the DeVIAS Letters in the hopes of finding a reference to a similar problem, or some utilities or tools to help me isolate our problem. I came up empty, so I had to obtain my own solution.

Our hardware consists of a 512K word PDP-11/70, with 1 RP06 and 2 RP04 drives. We are running a hybrid IAS V3.0, IAS V3.1 system. By this, I mean that the operating system proper is IAS V3.0, but we have taken some of the IAS V3.1 system utilities and run them on our system. Among the utilities we brought over are: EDT, DB.... handler, BRU. Our primary use of BRU is for disk compressions. We use the Bell Northern FBS product for our incremental backups.

FBS does not backup the FCS index file, or file directories. I needed to access the information in these files to determine if we had a file system structure problem. If a file was good one day, and went bad the next, then examination of the index and directory files would at least provide a starting point. One of our RP04's is a mountable device, so we designated a pack as an archive medium. Each morning, when our normal incremental backups were performed, copies of the index and directory ([0,0]INDEXF.SYS and [0,0]*.DIR;l) files were made in a private directory on the archive medium. Each of these private directories defined the state of the file system at the time the copies were made. Thus I now had time ordered descriptions of the file system structure.

I wrote a program, \$\$\$VRY, which allowed me to examine the archived states, as well as the current state, of the file system. The program examines all files in every directory, and checks their index file entries to ensure that there are no problems.

What I found out was that a very LARGE number of active file headers were not allocated in the index file bitmap! These file header numbers were widely scattered - there was no pattern that I could observe. This frightening situation accounted for the number of file-id, sequence number errors that we had. File headers which the index file believed were available for use, were actually active headers for existing files. If a new file is created, or extended, one of these "active" file headers can be allocated to the new file, since the index file bitmap says that the file header is available. As the "active" file header is allocated to the new file, the sequence number of the file header is bumped up, and BANG - the original file has a file-id, sequence number check failure. The FllACP does not check to see if an existing file's file header is allocated in the index file bitmap when it is asked to access the file, so we ended up with a slumbering time bomb.

We treated the symptoms by BRUing the disk in question to a new pack, then running VFY to ensure that the physical blocks were mapped ok, and then running \$\$\$VRY to ensure that the BRUed file system structure was ok. As to the disease itself, I have not determined what caused this problem. ANY SUGGESTIONS, OR THEORIES, WOULD BE WELCOME. Until we resolve the problem, we continue to make archival copies of the state of the file system, and we periodically run VFY and \$\$\$VRY on our media. Although we have not determined the disease, we can at least identify, and rectify, any trouble spots that need attention by use of \$\$\$VRY.

I have enclosed a tape (1600bpi, FILES-11 format) with the sources of \$\$\$VRY, and a brief summary of the utility. \$\$\$VRY is non-proprietary, and is in the public domain. Please inform DeVIAS members of the availability of this software - someone else may find it useful.

Sincerely;

John Gindi

This UFD contains the sources for the \$\$\$VRY file system verification program. The program as written works on IAS V3.0, and requires the library PIPUTL.OLB (assumed to be in [11,5]). This library is available on IAS systems from the IAS V3.0 system distribution, second object tape in UFD [11,5].

The utility is built using the indirect command file:

```
PDS> @VRY.CMD
```

The VRY.TSK image which is left behind must then be installed, normally as \$\$\$VRY.

Input to this utility consists of specifying the device, and the location where (possibly copies of) the FCS index file, and the directory files are located. The program reads each record in each directory file, and then examines the corresponding entries in the index file for consistency checks. Only the directories, and the index file in the location you specify are examined.

If there are any inconsistencies, a dump of the information of the directory entry, and the index file entries is made. Output may be redirected to a file, or device. By default, output is sent to the terminal. See the comments in the module VRYVFY.MAC to decipher the output.

This program can help detect the following situations: file-id, sequence number errors; directory entries for which there is no longer a file; active file headers which are about to be clobbered because the index file bitmap has not allocated them; inconsistency in multi-header files.

Either of the following command lines displays a help frame which describes the syntax and effect of this utility:

```
VRY> ?  
VRY> HELP
```

The syntax of the normal command line is:

```
VRY> [listingfile=] device:[ufd]
```

listingfile is a device, or file, for the output from this utility. device:[ufd] is the location of the index file and directory files you wish to verify.

Example usage: You suspect that something is attacking your file system. On Monday, you copy the current index file and directory files into a new directory to archive them for later use:

```
PDS> CREATE/DIR DB1:[77,1]  
PDS> COPY DB0:[0,0]*.DIR;l DB1:[77,1]  
PDS> COPY DB0:[0,0]INDEXF.SYS DB1:[77,1]
```

On Tuesday, you examine the archived status of the file system, as well as your active file system to determine if anything is amiss:

```
PDS> VRY DB1:[77,1]  
PDS> VRY DB0:[0,0]
```

John Guidi
The Computing Service
The Jackson Laboratory
Bar Harbor, ME
04609
phone: (207)288-3371 X-391

DIRECTORY MMØ:
5-MAR-84 10:51

README.LST;1	8.	05-MAR-84	00:00
VRY.CMD;1	1.	05-MAR-84	00:00
VRY.TKB;1	2.	05-MAR-84	00:00
VRYCML.MAC;1	24.	05-MAR-84	00:00
VRYDIR.MAC;1	14.	05-MAR-84	00:00
VRYDRV.MAC;1	6.	05-MAR-84	00:00
VRYHLP.MAC;1	4.	05-MAR-84	00:00
VRYIMP.MAC;1	10.	05-MAR-84	00:00
VRYINX.MAC;1	10.	05-MAR-84	00:00
VRYPRE.MAC;1	4.	05-MAR-84	00:00
VRYTRP.MAC;1	9.	05-MAR-84	00:00
VRYVFY.MAC;1	35.	05-MAR-84	00:00

TOTAL OF 127./127. BLOCKS IN 12. FILES

A Generic PDP-11

The PDP-11 family of computers was introduced in 1970. Today the PDP-11 family of computers and compatible peripherals is virtually unlimited in variety, speed, and cost. The Micro/T-11 was the first single chip implementation of the PDP-11 instruction set; it is now available in sample quantities. The Micro/J-11 has of course already made the Micro/T-11 look pretty drab (the Micro/J-11 is a PDP-11/70 on a chip). Are all of the implementations of the PDP-11 architecture (11/03, 11/04, 11/23, 11/24, 11/34, 11/40, 11/44, 11/45, 11/70, Micro/T-11, Micro/J-11, etc.) the same? Not quite!

First it is important to note that most of the basic instructions do operate the same from one PDP-11 processor to another, and programs can be written that are totally portable within the PDP-11 family of processor. So the things that we are going to look at here are not to be viewed as incompatibilities as much as "optional processor enhancements". The architecture has been preserved as each new generation of PDP-11 products has enhanced the instruction set and speed of the processor. what are the "basic instructions" that form the PDP-11 instruction set common to all of these processors? The question is more easily answered by describing the "optional processor enhancements".

EIS (Extended Instruction Set) was the first significant processor enhancement. The Extended Instruction set is fixed point multiply,

divide and shift enhancement (using the general purpose registers). The EIS instructions are MUL, DIV, ASH, ASHC. These same instructions are supported by using an Extended Arithmetic Element (EAE), which is a UNIBUS peripheral.

A Floating Point Processor (FPP) is an optional enhancement (11/34, 11/60, 11/45, 11/70) that provides a floating-point arithmetic instruction set with six associated registers.

The Floating-point Instruction Set (FIS) is an enhancement that enables a stack oriented floating-point instruction set (11/03, 11/35, 11/40).

Memory management is a feature that allows the use of a memory mapping unit for extended memory utilization (not implemented in 11/03, 11/04, 11/05, 11/10 and Micro/T-11).

If we eliminate the above enhancements we are still left with a 16-bit processor with a very comprehensive instruction set. All processors have R0-R5 as general purpose registers, SP (stack pointer), PSW (Processor Status Word), PC (Program Counter) and a common set of about 80 instructions that support various addressing modes.

Unfortunately, we still have some minor differences from processor to processor within this common PDP-11 instruction set. Usually these differences involve variations in the effects of execution of

instructions (condition code effected differently, autoincrement/ autodecrement acts differently, etc.) that are used very seldomly, and in fact should usually be avoided. An example of this is

```
MOV      R1,(R1)+
```

(same register in source and destination). Some processors (T11, 11/23, 11/15, 11/20, 11/35, 11/40) will increment R1 by two before being used as the source operand. Other processors do not increment R1 before it is used as a source operand!

Another example of the variations between processors is seen in instructions that involve the PC. The instruction

```
LABEL:   MOV      PC,PCSAVE
```

will either save LABEL+2 or LABEL+4 depending on the processor used - interesting huh? On a couple of processors a SWAB does not change the overflow flag in the PSW (most processors clear it). A

```
JMP      (R1)+
```

rarely (but occasionally) increments R1 before the jump.

Traps are implemented slightly differently on the various processors. In particular, the infamous "odd address or other trap 4"

is always "another trap 4" on an 11/24 - it doesn't trap odd addresses!

The I/O Page addresses for registers (177700-177717) are sometimes valid program addresses, sometimes only accessible from the console, and with the Micro/T-11 addressed as normal memory.

Power fail during and `RESET` instruction lets not get into that one!

The T-bit in the PSW may totally disallow interrupts (Micro/T-11), may allow the interrupt to take precedence, or may acknowledge the T-bit traps before the interrupt.

Stack pointer operations with an odd address reference sometimes causes a HALT, sometimes causes the creation of a new stack, or sometimes does not trap at all! Stack overflow detection is implemented on some but not all machines.

If a nonexistent memory error occurs using an auto-increment addressing mode some processors will increment the register, and others won't.

A HALT instruction executed in user mode may vector to trap 4 or trap 10 depending on processor type. On the Micro/T-11 a restart address is fetched and execution continues (with a PSW=340).

With all of the above mentioned differences (and many more minor differences), how can we possibly design "generic PDP/11 software"? The answer is twofold. First, we can carefully avoid all of the instruction constructs that tend to cause problems -- most of these are of limited use, and tend to be confusing anyway. Secondly, in those instances where we can't avoid the problem, the code should be written with conditional assembly directives that serves to both document the problem, and allow easy selection of code generation for the target processor.

However there is another possible approach. Determine which processor you are running on, and incorporate appropriate logic to handle the particular situation. The MFPT (move From Processor Type) instruction allows you to do exactly that. Each processor implemented (Micro/T-11, 11/23, 11/24, 11/44) has a unique code associated with it, and it is moved into R0 upon execution of the MFPT instruction. Without this instruction more exotic methods can be employed to determine the processor type, but it is much easier with MFPT.

Considering all of the above, we still have to conclude that any PDP-11 used with a common sense approach to coding can still be considered a "generic PDP-11". We simply have to be aware of the things to avoid, and write clean straightforward code.

Daniel Sedlacek
Informatics General Corp.
Bellevue NE
(402) 291-8300

12 Feb 84

John Drummond

Bob Curley

Here is a very late submission to the DEVIAS newsletter. I didn't have time to retype the attached. If it's unfit for print, please tell me and I'll get it ready for the next issue.

The utility is SRX, an IAS SRD. Basically, Bob Schult, the author of SRX, has cleaned up SRD and provided some more functionality. Bob Schuldt currently lives in Stuttgart Germany and has written a large number of utilities of HQ EUCOM (European Command). He will be retiring from the military soon and will probably be moving to Belgium.

Bob has also come up with a very, very nice snap dump type utility that provides as much or more information than a system crash dump. I don't have a copy of that one yet. I'm sure he will forward a copy to you, if it gets to me first I will send an overview up your way.

Ron Fussell

COI-SSUF11001

SRX COMPUTER OPERATING INSTRUCTIONS

PROCESSOR: PDP 11/70

OP SYSTEM: IAS V3.1

SUBSYSTEM: SYSTEM SOFTWARE UTILITIES

DOCUMENTATION REFERENCES:

IAS MCR USER'S GUIDE
IAS/RSX I/O OPERATIONS REFERENCE MANUAL
PDP-11 SORT REFERENCE MANUAL

FUNCTION:

SRX IS A FILES-11 DIRECTORY MANIPULATION UTILITY TASK WHICH PROVIDES THE USER WITH THE FOLLOWING CAPABILITIES:

- + SORTING OF A DIRECTORY INTO A SPECIFIED SEQUENCE FOR DISPLAY AND/OR DIRECTORY UPDATE PURPOSES
- + COMPRESSION OF A DIRECTORY FOR WRITEBACK
- + OUTPUT DIRECTORY LIST TO THE 'TI'
- + OUTPUT DIRECTORY LIST TO A DISK FILE WITH PAGING AND HEADINGS
- + CREATING AN OUTPUT DISK FILE CONTAINING COMMAND LINES
- + MASS SORTFILE INPUT/OUTPUT WITH 'SORT' INTERFACE
- + INPUT FILE SELECTION + DIRECTORY FILE MAINTENANCE

HARDWARE REQUIREMENTS:

- + 26KW MEMORY
- + 512-12000 BLOCKS OF DISK (BASED UPON USER REQUIREMENTS)

PREPARATION INSTRUCTIONS:

- + INSTALL [141,3]F11SRX
- + INSTALL [1,2]SORT
- + THE INPUT DISK MUST BE MOUNTED '/OVR', WITH WRITE ACCESS IF THE '/WB' OPTION IS UTILIZED OR AN OUTPUT FILE IS TO BE CREATED

GENERAL INFORMATION AND NOTES

1. SRX'S COMMAND FILE OUTPUT FUNCTION PROVIDES THE USER WITH AN EXTREMELY FLEXIBLE TOOL. THIS FUNCTION MAY BE USED TO CREATE A FILE CONTAINING EITHER DIRECTLY EXECUTABLE COMMAND LINE ENTRIES OR A FILE WHICH MAY BE EDITED TO INCLUDE INDIRECT COMMAND FILE PROCESSOR CONTROL STRINGS. THUS, THE OUTPUT COMMAND LINE FORMAT IS BASED ENTIRELY ON THE USER'S REQUIREMENTS. THE USER MAY ELECT TO CONCATENATE SEVERAL OF THE OUTPUT COMMAND FILES CREATED VIA SRX INTO A SINGLE FILE WHICH MAY THEN BE USED TO CONTROL A SPECIFIC PROCESS OR FUNCTION.

2. USAGE OF THE DIRECTORY WRITEBACK OPTION WILL BE AUTOMATICALLY CANCELLED IF ANY ONE OF THE FOLLOWING CONDITIONS IS ENCOUNTERED:

- A. ANY TYPE OF 'INFILE' SELECTION HAS BEEN REQUESTED.
- B. THE USER FILE DIRECTORY (UFD) IS TOO LARGE FOR SRX'S BUFFER.
- C. A PROBLEM WAS ENCOUNTERED WHILE LOADING THE DIRECTORY INTO MEMORY, (FCS RETURNED ANYTHING OTHER THAN A SUCCESS CODE).

IN THE EVENT THAT THE WRITEBACK OPTION WAS REQUESTED AND THE OPTION WAS CANCELLED, A MESSAGE WILL BE DISPLAYED NOTING THE CANCELLATION.

3. SRX MUST READ THE ENTIRE DIRECTORY INTO MEMORY BEFORE IT PERFORMS THE FUNCTIONS REQUESTED BY THE USER. IF SRX CANNOT LOAD THE ENTIRE DIRECTORY INTO MEMORY IN ONE PASS, SRX WILL COMPRESS THE CURRENT SECTION BY REMOVING UNUSED DIRECTORY ENTRIES, THEN, IF THERE IS ROOM FOR AT LEAST ONE BLOCK (512 BYTES (32 DIRECTORY ENTRIES)), SRX WILL LOAD ANOTHER BLOCK. SRX WILL CONTINUE THIS PROCESS UNTIL THERE IS INSUFFICIENT ROOM LEFT IN THE BUFFER TO LOAD ANOTHER BLOCK OR UNTIL THE ENTIRE DIRECTORY IS LOADED. WHEN SRX DETERMINES THAT IT CANNOT LOAD THE ENTIRE DIRECTORY, A WARNING MESSAGE WILL BE DISPLAYED AND THE DIRECTORY WILL BE PROCESSED IN CHUNKS.

4. IN THE EVENT THAT THE DIRECTORY IS TOO LARGE FOR SINGLE SHOT PROCESSING (MORE THAN 2048 ENTRIES), THE USER MAY WISH TO SELECTIVELY PURGE/DELETE DIRECTORY ENTRIES. THIS MAY BE ACCOMPLISHED VIA THE MAINTENANCE SWITCHES ('/SD', '/DE' OR '/PU'). ONCE THE NUMBER OF DIRECTORY ENTRIES HAVE BEEN DECREASED TO THE RANGE OF 2016-2048, SRX WILL ALLOW THE DIRECTORY TO BE PROCESSED IN ONE SHOT.

5. SRX CONTAINS A SPECIAL ROUTINE FOR HANDLING THE MASTER FILE DIRECTORY (MFD, [0,0]). ANYTIME THE MFD IS PROCESSED, THE FIRST FIVE FILES (CREATED VIA THE VOLUME INITIALIZATION PROCESS TASK '..IN1'), WILL BE LOCATED AND LOADED IN ORDER AS THE FIRST FIVE DIRECTORY ENTRIES. THIS CORRECTS PROBLEMS CAUSED BY PREVIOUS DIRECTORY MANIPULATION TASKS (IE: '..SRD' AND '..DIR') WHICH RESULTED IN SPURIOUS ERROR MESSAGES BEING OUTPUT BY THE FILES-11 DISK VOLUME VALIDATION TASK '..VFY'. IN ADDITION, SRX WILL NOT ALLOW THE USER TO USE THE '/DE' OPTION WHEN PROCESSING THE MFD.

COI-SSUF11001

6. SRX SHOULD ALLOW BE EXECUTED VIA A HARDCOPY TERMINAL WHEN PERFORMING CONFIGURATION CONTROL FUNCTIONS TO ENABLE A PERMANENT RECORD OF THE ACTIONS TAKEN.

7. ALL OUTPUT LIST FILES CREATED BY SRX ARE PAGED AT 59(10) LINES PER PAGE. THE FIRST FOUR LINES OF EACH PAGE CONSIST THREE PAGE HEADER LINES FOLLOWED BY A BLANK LINE. THE FIRST PAGE HEADING LINE CONTAINS THE DEVICE, UIC, UFD ACCESS TIME, INPUT COMMAND LINE AND THE PAGE NUMBER. LINES TWO AND THREE IDENTIFY THE DATA FIELDS. IF THE USER DID NOT REQUEST A PAGE EJECT '/EJ' ON CHANGE IN UIC, AND THE CURRENT PAGE HAS AT LEAST SIX LINES REMAINING, (THIS ALLOWS FOR THE NEW UIC TO BE IDENTIFIED AND ONE DIRECTORY ENTRY TO BE DISPLAYED), A PAGE HEADING WILL INSERTED WITH A PAGE NUMBER, IF NOT, A NEW PAGE WILL BE STARTED.

8. UFD'S WHICH CONTAIN NO DIRECTORY ENTRIES WILL NOT BE DISPLAYED DURING LIST OPERATIONS, THEY WILL BE CONTAINED IN THE 'NULL DIRECTORY' SECTION OF THE FINAL TOTALS (/TB).

9. WHEN A PROBLEM IS ENCOUNTERED BY SRX PROCESSING THE FILE-HEADER FOR A DIRECTORY ENTRY, SRX WILL DISPLAY A TWO LINE ERROR MESSAGE IDENTIFYING THE FILE AND THE TYPE OF ERROR. THIS MESSAGE WILL BE DISPLAYED WITH A LEADING AND TRAILING BLANK LINE. IF AN OUTPUT LIST FILE IS BEING CREATED, THE ERROR MESSAGE WILL BE DISPLAYED ON THE INPUT TERMINAL AND RECORDED IN THE OUTPUT LIST FILE.

10. DUE TO THE FACT THAT SRX MAY BE USED TO LIST AND/OR MAINTAIN A UFD, SOME TRACE OF SRX'S USAGE WILL BE RECORDED EITHER ON THE INPUT TERMINAL OR BY THE FACT THAT AN OUTPUT FILE OF SOME TYPE HAS BEEN CREATED. SRX IDENTIFIES EACH FILE THAT IT CREATES ALONG WITH THE NUMBER OF PAGES CONTAINED IN THE OUTPUT LIST FILE OR THE NUMBER OF ENTRIES IN A COMMAND/SORT FILE. SRX WILL NOT ALLOW THE USER TO ENTIRELY SHUTDOWN ITS OUTPUT. IF THE USER CANCELS THE DEFAULT SWITCHES '/LI/TB' BY ENTERING '/-LI/-TB', SRX WILL AUTOMATICALLY GENERATE A '/BR' FORM OUTPUT.

11. ALL FILES ARE REVISED AT LEAST ONE TIME. THIS OCCURS WHEN THE FILE IS CLOSED AND THE FILE HEADER IS UPDATED WITH THE END-OF-FILE INFORMATION. CONSEQUENTLY, FILE REVISION DATA WILL ONLY BE DISPLAYED (WITH THE '/FU' OR '/LS' SWITCHES) WHEN THE FILE HAS BEEN REVISED TWO OR MORE TIMES.

COI-SSUF11001

CONSOLE INPUTS:

SRX IS DRIVEN BY THE COMMAND LINES INPUT VIA THE CONSOLE OR AN INDIRECT COMMAND FILE. THE COMMAND LINE FORMAT IS DEFINED AS FOLLOWS:

>SRX [OUTFILE[/SWITCHES]=][INFILE][/SWITCHES][,SORTFILE]

NOTE THAT ALL FIELDS ARE OPTIONAL. THE 'OUTFILE', 'INFILE' AND 'SORTFILE' ARE ANY FILE SPECIFICATIONS IN STANDARD IAS/RSX FORMAT. THE DEFAULT COMMAND LINE AND ITS DEFAULT VALUES ARE DEFINED AS FOLLOWS:

>SRX TI:(LOGON UIC)UFD.LST/-SP=SY0:(LOGON UIC)*.*;*/LI/TB/SR:ANATAV

	"OUTFILE"	"INFILE"	"SORTFILE"	
	-----	-----	-(OUTPUT)-	-(INPUT)-
DEVICE	SY0:	SY0:	SY0:	SY0:
UIC	LOGON	LOGON	LOGON	LOGON
FILENAME	UFD	*	SRX	SRX
FILETYPE	.LST	.*	.DAT	.DAT
VERSION	+1	;*	+1	;0
SWITCHES	/-SP	/LI/TB /SR:ANATAV		

EXCEPT THAT IF NO 'OUTFILE' SPECIFICATION IS GIVEN, THE OUTPUT DEVICE WILL BE DEFAULTED TO THE 'TI'.

WILDCARDS (*) MAY BE USED IN THE 'INFILE' UIC SPECIFICATION. FOR EXAMPLE THE COMMAND

>SRX [1,*]

LISTS ON THE INPUT TERMINAL ALL FILE ENTRIES IN ALL USER FILE DIRECTORIES (UFD'S) THAT BELONG TO GROUP 1. UFD'S CONTAINING NO ACTIVE ENTRIES WILL NOT BE LISTED, INSTEAD THE UIC WILL BE LISTED IN THE 'NULL DIRECTORIES' SECTION OF THE FINAL TOTALS (/TB).

THE 'FILENAME', 'FILETYPE' AND 'VERSION' FIELDS OF THE 'INFILE' SPECIFICATION MAY BE USED TO SELECT SPECIFIC SETS OF FILES. THE WILDCARD (*) MAY BE USED IN ANY ONE OF THESE FIELDS. FOR EXAMPLE, THE COMMAND

>SRX *.CMD

WILL LIST ON THE INPUT TERMINAL ALL FILES IN THE LOGON UIC WITH A FILETYPE OF ".CMD".

FILES MAY ALSO BE SELECTED BY USING WILD CHARACTERS IN THE 'FILENAME' AND 'FILETYPE' FIELDS OF THE 'INFILE' SPECIFICATION. THE WILD CHARACTERS WHICH MAY BE USED AND THERE MEANING ARE DEFINED AS FOLLOWS:

CHARACTER	MEANING
-----	-----
*	MATCH ZERO OR MORE CHARACTERS IN THIS FIELD
-	MATCH ONE CHARACTER IN THIS FIELD
?	(SAME AS '-')
%	(SAME AS '-')

COI-SSUF11001

FOR EXAMPLE, THE COMMAND

>SRX EU*

WILL LIST ON THE INPUT TERMINAL ALL FILE ENTRIES IN THE LOGON UIC WITH "EU" AS THE FIRST TWO CHARACTERS OF THE FILENAME. IN CONTRAST, THE "EU" FILENAME SPECIFICATION MATCHES ONLY THOSE FILES WHICH HAVE A FILENAME OF "EU". THUS, THE PATTERN "EU*" MATCHES A FILE WITH THE FILENAME OF "EUSSGEN", WHILE THE PATTERN "EU" DOES NOT. NOTE THAT THE FILETYPE AND VERSION FIELDS ARE WILDCARDS BY DEFAULT.

>SRX EU---G*

THIS COMMAND SPECIFIES THAT ALL FILES WITH "EU" IN POSITIONS ONE AND TWO, HAVE ANY CHARACTERS IN POSITIONS THREE, FOUR AND FIVE, AND HAVE A "G" IN THE SIXTH POSITION ARE TO BE SELECTED. THE WILDCARD (*) MATCHES EVERYTHING. THE WILDCARD (*) MAY BE USED ANYWHERE IN THE FILENAME AND FILETYPE FIELDS, AT THE BEGINNING, INTERMEDIATE OR END POSITIONS. FOR EXAMPLE:

>SRX S*D?T

SPECIFIES ALL FILES WITH A FILENAME STARTING WITH AN "S", HAVING ANY NUMBER OF CHARACTERS (INCLUDING ZERO), AND THEN HAVE A "D" FOLLOWED BY ANY SINGLE CHARACTER, AND THEN A "T" ARE TO BE SELECTED. THE RESULT IS THE SAME AS A SINGLE-COMMAND COMBINATION OF THE COMMANDS:

>SRX SD-T
>SRX S-D-T
>SRX S--D-T
>SRX S---D-T
>SRX S----D-T
>SRX S-----D-T

ANOTHER EXAMPLE IS:

>SRX *MAC*

THIS COMMAND WOULD CAUSE ALL FILES CONTAINING THE PATTERN "MAC" ANYWHERE IN THE FILENAME TO BE SELECTED, SUCH AS

MAC,BAS
PURMAC,TSK
MACMAC,BAS
MACSLP,CMD
USAIPGMAC,CMD
AMAC,BAS
MACALL,CMD

IF THE VERSION IS SPECIFIED, IT HAS THE USUAL MEANING (DEFAULT = *)

-1 SELECT OLDEST VERSION
0 SELECT LATEST VERSION
* SELECT ALL VERSIONS
'N' SELECT THE SPECIFIED VERSION

ANY OF THE SWITCH OPTIONS MAY BE USED FOLLOWING THE 'OUTFILE', 'INFILE' OR 'SORTFILE' SPECIFICATIONS. THE VALID SWITCH COMBINATIONS AND FILE SPECIFICATION APPLICABILITY ARE DEFINED IN THE FOLLOWING TABLE.

	A	B	C	D	E	F	H	L	L	M	N	O	P	R	S	S	S	S	T	W	X	Z
INFILE	X	X		X	X	X		X	X	X	X	X	X	X	X	X			X	X	X	
OUTFILE			X	X			X	X	X										X	X	X	
SORTFILE	X	X		X	X		X	X	X	X	X	X	X	X	X			X			X	X
/AF				<																	C	.
/BE				<																	C	.
/BR										N					N							
/CF			N			N	N	N		N			I								N	N
/CO																						C
/DA																						C
/DE				N				N	N			I										C
/EJ										N					N				N	N		
/FU				<	N										N				N	N		
/HV														I	I							C
/LI				<	I					N												
/LO																						C
/LS				N	I			I	N	N		N									I	I
/MU																						C
/NE																						C
/NV								I						I	I							C
/OV								I						I	I							I
/PU				N	I			I	N	N	I			I								C
/RD																						C
/SD				N	I			I	N	N	I			I								C
/SF								N	N		N			I								
/SP																						
/SR											I											
/TB					N																	
/WB		I	I		I	I	I		I	I	I	I	I	I	I	I	I					I
/XN				N																		
/ZE																						C

C = SWITCH CANCELLED X = SWITCH USED FOR FILE SPECIFICATION
I = INCOMPATIBLE < = OVERRIDES ^ = OVERRIDES <
N = NON-APPLICABLE ^ = OVERRIDES <

SWITCHES FALL INTO THE FOLLOWING FOUR CATEGORIES.

- OUTPUT FORM /BR, /LI, /FU, /TB, /EJ, /XN, /SP, /CF
- SELECTION /AF, /BE, /DA, /RD, /CO, /LO, /MU, /ZE, /HV, /NV, /OV, /NE
- SORT /SR, /SF, /LS
- MAINTENANCE /SD, /DE, /PU, /WB

COI-SSUF11001

CONSOLE HEADING FORMAT:

LNE	LOC	LEN	DATA TYPE	CONTENTS/COMMENTS
1	001	005	ASCII	INPUT DIRECTORY DEVICE 'DDNN:'
	006	009	ASCII	INPUT USER FILE DIRECTORY (UFD) '[G,M]'
	015	002		SPACER
	017	018	FMTD DATE	UFD ACCESS TIME 'DD-MMM-YY HH:MM:SS'
2	001	080	ASCII	INPUT COMMAND LINE

OUTPUT LIST FILE HEADING FORMAT:

LNE	LOC	LEN	DATA TYPE	CONTENTS/COMMENTS
1	001	005	ASCII	INPUT DIRECTORY DEVICE 'DDNN:'
	006	009	ASCII	INPUT USER FILE DIRECTORY (UFD) '[G,M]'
	015	002		SPACER
	017	018	FMTD DATE	UFD ACCESS TIME 'DD-MMM-YY HH:MM:SS'
	035	003		SPACER
	038	080	ASCII	INPUT COMMAND LINE
	118	003		SPACER
	121	005	CONSTANT	'PAGE:'
	126	005	DECIMAL	PAGE NUMBER
2	001	130	CONSTANT	FIELD NAMES
3	001	130	CONSTANT	UNDERLINES

OUTPUT LIST FILE DIRECTORY ENTRY DISPLAY FORMAT:

XRF	LOC	LEN	DATA TYPE	CONTENTS/COMMENTS
AA	001	020	ASCII	FILENAME (TYPE AND VERSION)
	021	001		SPACER
AB	022	001	FLAG	'C' IF FILE IS CONTIGUOUS
AC	023	001	FLAG	'L' IF FILE IS LOCKED
AD	024	001	FLAG	'M' IF THE FILE HAS MUTIPLE FILE-HEADERS
	025	001		SPACER
AE	026	018	FMTD DATE	CREATION DATE AND TIME 'DD-MMM-YY HH:MM:SS'
	044	001		SPACER
AF	045	006	DECIMAL	NUMBER OF BLOCKS USED
	051	001		SPACER
AG	052	006	DECIMAL	NUMBER OF BLOCKS ALLOCATED
	058	001		SPACER
AH	059	013	FMTD	FILE-IDENTIFICATION '(NNNNN,SSSS)', WHERE
				'N' = FILE NUMBER AND 'S' = SEQUENCE NUMBER
	072	001		SPACER
AI	073	026	FMTD	REVISION DATA, 'DD-MMM-YY HH:MM:SS (N.)'
	099	001		SPACER
AJ	100	009	FMTD	FILE OWNER '[G,M]' ('/LS' = FILE UIC)
	109	001		SPACER
AK	110	021	FMTD	FILE PROTECTIONS '[RWED,RWED,RWED,RWED]'

COI-SSUF11001

OUTPUT LIST FILE DIRECTORY ENTRY DISPLAY FORMAT:

THE TERMINAL DISPLAY FORMAT IS THE SAME AS THE LIST FILE FORMAT EXCEPT THAT A TWO POSITION FIELD CONTAINING SPACES WILL BE INSERTED BEFORE THE FILENAME FIELD (XFR AA).

SELECTION SWITCH CROSS-REFERENCE TABLE

REF	SWITCHES
AA	/NE, /HV, /NV, /OV, /PU
AB	/CO
AC	/LO
AD	/MU
AE	/DA, /BE, /AF
AF	/ZE
AG	
AH	
AI	/RD, (/DA, /BE, /AF)
AJ	
AK	

DISPLAY SWITCH CROSS-REFERENCE TABLE

REF	/BR	/LI	/FU	/DE	/SD	/PU	/LS
AA	X	X	X	X	X	X	X
AB		X	X	X	X	X	X
AC		X	X	X	X	X	X
AD		X	X	X	X	X	X
AE		X	X	X	X	X	X
AF		X	X	X	X	X	X
AG		X	X	X	X	X	X
AH			X				X
AI			X				X
AJ			X				*1
AK			X				X
*2					X	X	

*1 = UIC OF FILE IS DISPLAYED VICE FILE OWNER
*2 = SELECTIVE DELETE PROMPT '[Y/N/G/Q]? _'

COI-SSUF11001

SORT FILE ENTRY FORMAT:

XRF	LOC	LEN	DATA TYPE	CONTENTS/COMMENTS
---	---	---	-----	-----
BA	001	002	BINARY	FILE NUMBER
BB	003	002	BINARY	FILE SEQUENCE NUMBER
BC	005	002	RAD50	USER SUPPLIED PACK-ID (/SF:XXX)
BD	007	006	RAD50	FILENAME
BE	013	002	RAD50	FILETYPE
BF	015	002	BINARY	VERSION NUMBER
BG	017	002	BINARY	CREATION DATE (MONOTOMIC)
BH	019	003	PACK-DEC	CREATION TIME (HHMMSS)
BI	022	001	BINARY	'CLM' FLAGS
BJ	023	002	BINARY	BLOCKS USED
BK	025	002	BINARY	BLOCKS ALLOCATED
BL	027	002	BINARY	FILE PROTECTIONS
BM	029	002	BINARY	FILE OWNER
BN	031	002	BINARY	UIC
BO	033	002	BINARY	REVISION COUNT
BP	035	002	BINARY	REVISION DATE (MONOTOMIC)
BQ	037	003	PACK-DEC	REVISION TIME (HHMMSS)
BR	040	001	NULL	WORD-ALIGNMENT PAD
BS	041	009	ASCII	FILENAME
BT	050	003	ASCII	FILETYPE
BU	053	002	BINARY	VERSION NUMBER
BV	055	003	OCTAL	UIC GROUP CODE
BW	058	003	OCTAL	UIC MEMBER CODE

SORT FILE SORT SEQUENCE:

XRF	LOC	LEN	DATA TYPE	COMMENTS
---	---	---	-----	-----
BS	041	012	ASCII	'CN41.12' - MAJOR
BN	031	002	BINARY	'BN31.2'
BW	044	006	ASCII	'CN44.6' - MINOR

COI-SSUF11001

OUTPUT FORM SWITCHES

>> SPOOL OUTPUT /SP

THIS SWITCH CAUSES SRX TO SPOOL THE OUTPUT LIST FILE FOR PRINTING. THE DEFAULT IS /-SP.

>> TABULAR SUMMARY /TB

THIS SWITCH IS USED TO DISPLAY SUMMARY TOTALS ACCUMULATED WHILE PROCESSING A UIC. THE DEFAULT IS '/TB'. THE TOTALS ARE DISPLAYED IN THE FOLLOWING FORMAT:

TOTAL FILES - SELECTED: (N1). INPUT: (N2).
TOTAL BLOCKS USED/ALLOCATED: (N3)/(N4).

WHERE: N1 IS THE TOTAL NUMBER OF FILES SELECTED FOR OUTPUT
N2 IS THE TOTAL NUMBER OF FILE ENTRIES CONTAINED IN THE UFD
N3 IS THE TOTAL NUMBER OF BLOCKS IN USE BY THE SELECTED FILES
N4 IS THE TOTAL NUMBER OF BLOCKS ALLOCATED TO THE SELECTED FILES

IN ADDITION, WHEN THE USER USES THE WILDCARD DIRECTORY OPTION, SRX WILL OUTPUT A FINAL TOTAL SUMMARY. THIS SUMMARY IS THE ACCUMULATION OF ALL THE DIRECTORIES PROCESSED FOR THIS REQUEST AND IS DISPLAYED IN THE FOLLOWING FORMAT:

--- FINAL TOTALS -

--- NULL DIRECTORIES -

(F1)

TOTAL FILES - SELECTED: (F2). INPUT: (F3). UFD'S: (F4).
TOTAL BLOCKS USED/ALLOCATED: (F5)/(F6).

WHERE: F1 WILL BE A STRING OF UIC'S WHICH WERE PROCESSED AND WHERE FOUND TO HAVE NO DIRECTORY ENTRIES
F2 IS THE SUM OF ALL PREVIOUS N1 TOTALS
F3 IS THE SUM OF ALL PREVIOUS N2 TOTALS
F4 IS THE TOTAL NUMBER OF DIRECTORIES PROCESSED
F5 IS THE SUM OF ALL PREVIOUS N3 TOTALS
F6 IS THE SUM OF ALL PREVIOUS N4 TOTALS

>> PAGE EJECT /EJ

THIS SWITCH CAUSES A NEW PAGE TO BE GENERATED IN THE OUTPUT LIST FILE WHEN A CHANGE IN UIC OCCURS. THE DEFAULT IS /-EJ.

COI-SSUF11001

>> EXPAND FILENAME /XN

THIS SWITCH CAUSES THE FILENAME (TYPE AND VERSION) TO BE DISPLAYED IN THE FIXED FORMAT 'NNNNNNNNN.TTT;VVVVVV'. THE DEFAULT IS '/-XN'. USE OF THIS SWITCH INCREASES THE READABILITY OF THE OUTPUT DISPLAY.

>> BRIEF DIRECTORY LIST /BR/-LI

THESE SWITCHES CAUSE THE OUTPUT DIRECTORY DISPLAY TO CONTAIN ONLY THE FILENAME (XRF AA). SINCE THE DEFAULT LIST SWITCH IS '/LI', TO ACHIEVE A BRIEF LIST, THE '/LI' SWITCH MUST BE NEGATED. THE DEFAULT IS '/-BR'.

>> MEDIUM DIRECTORY LIST /LI[:N]

THIS SWITCH CAUSES THE FILENAME THRU BLOCKS ALLOCATED DISPLAY FIELDS TO BE OUTPUT (XRF AA-AG). THE DEFAULT IS '/LI'. THE 'N' SUBFIELD WILL ALLOW THE USER TO TRUNCATED THE OUTPUT LINE LENGTH TO THE SPECIFIED VALUE OF 'N'. IN OTHER WORDS, IF THE USER WISHES TO PRODUCE AN OUTPUT LINE 76(10) CHARACTERS WIDE, THEY WOULD INPUT '/LI:76'.

>> FULL DIRECTORY LIST /FU[:N]

THIS SWITCH CAUSES THE FILENAME THRU FILE PROTECTIONS DISPLAY FIELDS TO BE OUTPUT (XRF AA-AK). THE DEFAULT IS '/-FU'. THE 'N' SUBFIELD HAS THE SAME MEANING AS DEFINED WITH THE '/LI' SWITCH.

>> CREATE COMMAND FILE /CF<TEMPLATE>

THIS SWITCH ALLOWS THE USER TO CREATE AN OUTPUT FILE WHICH WILL CONTAIN ONE COMMAND LINE FOR EACH FILE SELECTED. THE DEFAULT IS '/-CF'.

THE ACTUAL CONTENTS OF EACH OUTPUT COMMAND LINE IS DETERMINED BY THE COMMAND LINE BUILD TEMPLATE INPUT BY THE USER. THE OUTPUT FILE WILL NOT CONTAIN PAGE HEADINGS NOR WILL IT BE PAGED. THE PROCESSING OF THE TEMPLATE IS CONTROLLED BY THE FOLLOWING SPECIAL CHARACTERS:

CHAR USAGE

< START OF TEMPLATE, NOT INCLUDED IN OUTPUT COMMAND LINE
" REPLACE WITH THE ASCII UIC OF THE FILE
\$ REPLACE WITH THE 'FILENAME' OF THE FILE
! REPLACE WITH THE '.FILETYPE' OF THE FILE
? REPLACE WITH THE ';VERSION' OF THE FILE
> END-OF-TEMPLATE, NOT INCLUDED IN OUTPUT COMMAND LINE
'X' ANY OTHER CHARACTER WILL BE TRANSFERRED DIRECT

FOR EXAMPLE:

>SRX TMP.CMD=[11,1]*.TSK;0/CF<PIP 'TGT'*/FO/NV/CD='SRC' "\$! ?>

WOULD CREATE A FILE NAMED 'TMP.CMD' IN THE LOGON UIC CONTAINING ONE COMMAND LINE FOR THE LATEST VERSION OF EACH FILE CONTAINING A FILETYPE OF '.TSK'. THE COMMAND LINE WOULD APPEAR AS FOLLOWS:

PIP 'TGT' [11,1]/FO/NV/CD='SRC' [11,1](FILENAME).TSK;(VERSION)

COI-SSUF11001

THE OUTPUT COMMAND FILE MAY THEN BE EDITTED TO INCLUDE THE SUBSTITUTION
COMMAND LINES REQUIRED BY THE 'INDIRECT COMMAND FILE PROCESSOR' TO USE
THIS FILE AS AN EXECUTABLE INPUT.

FOR EXAMPLE:

```
.ASKS TGT WHAT IS THE TARGET DEVICE [IE: DBO:]?  
.ASKS SRC WHAT IS THE SOURCE DEVICE [IE: CMO:]?  
PIP 'TGT'[11,1]/FO/NV/CD='SRC'[11,1](FILENAME).TSK;(VERSION)  
PIP 'TGT'[11,1]/FO/NV/CD='SRC'[11,1](FILENAME).TSK;(VERSION)  
PIP 'TGT'[11,1]/FO/NV/CD='SRC'[11,1](FILENAME).TSK;(VERSION)  
ETC.
```

NOTE: IF WE HAD DEFAULTED THE VERSION FIELD BY NOT SPECIFYING IT, THEN
ONE OUTPUT COMMAND LINE WOULD HAVE GENERATED FOR EACH FILE FOUND WITH
A FILETYPE OF '.TSK'.

SELECTION FORM SWITCHES

- >> AFTER DATE /AF[:DD-MMM-YY[-E]] OR [:DDMMYY[-E]]

THIS SWITCH IS USED TO SELECT ALL FILES CREATED AFTER (PLUS ON IF '-E'
IS APPENDED) THE SPECIFIED DATE. IF NO DATE IS ENTERED, THE DEFAULT IS
THE CURRENT SYSTEM DATE. THE DEFAULT IS '/-AF'.
- >> BEFORE DATE /BE[:DD-MMM-YY[-E]] OR [:DDMMYY[-E]]

THIS SWITCH IS USED TO SELECT ALL FILES CREATED BEFORE (PLUS ON IF '-E'
IS APPENDED) THE SPECIFIED DATE. IF NO DATE IS ENTERED, THE DEFAULT IS
THE CURRENT SYSTEM DATE. THE DEFAULT IS '/-BE'.
- >> DATE /DA[:DD-MMM-YY] OR [:DDMMYY]

THIS SWITCH IS USED TO SELECT ALL FILES CREATED ON THE DATE SPECIFIED.
IF NO DATE IS SPECIFIED, THE DEFAULT IS THE CURRENT SYSTEM DATE.
- >> REVISION DATE /RD

THIS SWITCH CAUSES SRX TO USE THE REVISION DATE OF A FILE FOR SELECTION
VERSUS THE CREATION DATE. IF THERE IS NO REVISION DATE, THE CREATION
DATE WILL BE USED. THE DEFAULT IS '/-RD' (USE CREATION DATE).
- >> CONTIGUOUS FILE /CO

THIS SWITCH IS USED TO SELECT ONLY THOSE FILES WHICH ARE CONTIGUOUS.
- >> LOCKED FILE /LO

THIS SWITCH IS USED TO SELECT ONLY THOSE FILES WHICH ARE LOCKED.
- >> MULTIPLE-HEADERS /MU

THIS SWITCH IS USED TO SELECT ONLY THOSE FILES WHICH HAVE MULTIPLE
FILE-HEADERS.

COI-SSUF11001

>> ZERO BLOCKS USED /ZE

THIS SWITCH IS USED TO SELECT ONLY THOSE FILES WHICH HAVE ZERO BLOCKS USED.

>> NEGATE FILENAME LOGIC /NE

THIS SWITCH IS USED TO SELECT FILES WHICH DO NOT MATCH THE FILENAME SELECTION PARAMETERS. (SOMETIMES IT IS EASIER TO SPECIFY A SUBSET OF FILES NOT WANTED INSTEAD OF THE FILE WANTED).

NOTE THE NEXT FOUR SWITCHES ARE MUTUALLY EXCLUSIVE AND WILL BE OVERRIDEN IF THE VERSION NUMBER IS SPECIFIED AS PART OF THE FILENAME SELECTION PARAMETERS.

>> NEWEST VERSION /NV[:N]

THIS SWITCH CAUSES THE 'N' NEWEST VERSIONS OF A FILE TO BE SELECTED. IF 'N' IS NOT SPECIFIED, DEFAULT WILL BE THE LATEST VERSION.

>> OLDEST VERSION /OV[:N]

THIS SWITCH CAUSES THE 'N' OLDEST VERSIONS OF A FILE TO BE SELECTED. IF 'N' IS NOT SPECIFIED, DEFAULT WILL BE THE OLDEST VERSION.

>> HIGHEST VERSION /HV[:N]

THIS SWITCH CAUSES ALL VERSIONS OF A FILE THAT HAVE A VERSION NUMBER HIGHER THAN 'N'. IF 'N' IS NOT SPECIFIED, THE DEFAULT IS TWO (2).

SORT FORM SWITCHES

>> SORT UPD /SR[:F1F2F3]

THIS SWITCH CAUSES SRX TO SORT THE USER FILE DIRECTORY TO BE SORTED IN THE SEQUENCE SPECIFIED. THE DEFAULT IS '/SR:ANATAV' WHICH WILL SORT THE DIRECTORY INTO ASCENDING SEQUENCE BY THE FULL FILENAME.

THE SORT SEQUENCE IS CONTROLLED BY THE SUBSWITCH FIELDS 'F1', 'F2' AND 'F3'. THE FIRST CHARACTER OF THE SUBSWITCH FIELD DETERMINES WHETHER OR NOT THE SEQUENCE IS ASCENDING (A) OR DESCENDING (D). THE NEXT CHARACTER DETERMINES THE FIELD TO BE SORTED, FILENAME (N), TYPE (T) OR VERSION (V). IF THE USER INPUTS LESS THAN THREE SUBSWITCH FIELDS, THE UNUSED FIELDS WILL BE DEFAULTED TO ASCENDING SEQUENCE ON THE UNUSED ENTRIES IN THE DEFAULT ORDER OF FILENAME, TYPE AND VERSION.

FOR EXAMPLE:

>SRX (11,1)/SR:DT

WOULD SORT UIC (11,1) INTO THE FOLLOWING SEQUENCE:

DESCENDING	- FILETYPE	MAJOR
ASCENDING	- FILENAME	INTERMEDIATE
ASCENDING	- VERSION	MINOR

THERE ARE TWO EXCEPTIONS TO BE NOTED. IF THE USER SPECIFIED NO SORT ('/-SR'), THE SEQUENCE WILL BE DIFFER FROM THAT ACTUALLY CONTAINED IN THE DIRECTORY SINCE SRX WILL ALWAYS COMPRESS THE DIRECTORY PRIOR TO CHECKING FOR THE SORT SPECIFICATION. THE SECOND HAS TO DO WITH [0,0], THE MASTER FILE DIRECTORY (MFD). WHEN A VOLUME IS INITIALIZED, THERE ARE FIVE FILES WHICH ARE CREATED BY THE INITIALIZATION TASK '...INI'. THE FILES-11 VALIDATION TASK '...VFY' EXPECTS THE FIRST FIVE FILES TO BE FOUND AS THE FIRST FIVE ENTRIES IN THE MFD. IF THEY ARE NOT, AN ERROR MESSAGE WILL BE GENERATED. THUS, IF THE MFD IS BEING PROCESSED, SRX WILL SCAN THE DIRECTORY, LOCATE THE FIVE FILES (INDICATED BY THE FILE-ID NUMBER BEING THE SAME AS THE FILE-SEQUENCE NUMBER), THEN MOVE THEM TO THE CORRECT LOCATION PRIOR TO SORTING THE REST OF THE DIRECTORY INTO THE REQUESTED SEQUENCE.

>> CREATE SORT FILE /SF[:XXX]

THIS SWITCH CAUSES SRX TO CREATE A MASS SORT FILE CONTAINED ONE RECORD FOR EACH FILE SELECTED. THE SUBFIELD 'XXX' MAY BE USED TO INCLUDE AN IDENTITY FOR THE FILE. THIS VALUE WILL BE STORED IN EACH OUTPUT RECORD AND IS DISPLAYED AS THE 'PACK-ID' FIELD IN THE FIRST HEADING LINE OF EACH PAGE GENERATED BY THE MASS SORT FILE LIST OPTION ('/LS'). THE DEC PDP-11 SORT ('...SRT') TASK WILL BE USED TO SEQUENCE THE IAW WITH THE PARAGRAPH TITLED "SORT FILE SORT SEQUENCE".

>> LIST SORT FILE /LS[:N]

THIS SWITCH IS USED TO LIST A PREVIOUS SORT FILE CREATED BY SRX. THE FORMAT WILL BE A '/FU' OPTION WITH THE UIC OF THE FILE BEING DISPLAYED IN THE FILE OWNER FIELD. THE 'N' SUBFIELD ALLOWS THE USER TO SPECIFY WHETHER OR NOT BLANK LINES WILL BE INSERTED ON A CHANGE-OF-SEQUENCE. IF THIS FIELD IS OMITTED, NO CONTROL BREAK LINES WILL BE INSERTED. THE CONTROL BREAK SEQUENCE IS DEFINED AS FOLLOWS:

THE FILENAME AND TYPE FIELDS ARE TREATED AS ONE FIELD IN EXPANDED FORMAT ('FFFFFFFFTTT'). THE VALUE OF 'N' EQUATES TO ONE (1) FOR THE FIRST THREE CHARACTERS, TWO (2) FOR THE FIRST SIX CHARACTERS, THREE (3) FOR THE FIRST NINE CHARACTERS AND FOUR (4) FOR THE ENTIRE TWELVE CHARACTERS.

ONLY ONE LINE WILL BE INSERTED FOR EACH CONTROL BREAK, REGARDLESS OF WHAT SECTION CAUSED THE CONTROL BREAK (1, 2, 3 OR 4).

COI-SSUF11001

MAINTENANCE FORM SWITCHES

***** THESE SWITCHES SHOULD ONLY BE USED WHEN A HARD COPY (TERMINAL OR FILE)
***** OUTPUT IS BEING GENERATED. THIS IS TO INSURE AN AUDIT TRAIL WHICH WILL
***** IDENTIFY THE MAINTENANCE ACTION TAKEN.

>> SELECTIVE DELETE /SD

THIS SWITCH CAUSES SRX TO DISPLAY EACH FILE SELECTED WITH A PROMPT FOR ACTION ('[Y/N/G/Q]? _'). THE USER HAS THE FOLLOWING CHOICE OF ACTIONS:

Y = DELETE THE FILE
N = DISPLAY THE NEXT FILE
G = DELETE THE REST OF THE SELECTED FILES WITHOUT PROMPT, IF INPUT, THE USER WILL BE PROMPTED WITH THE FOLLOWING PROMPT:

ARE YOU SURE [Y/N]?

THE USER MUST ENTER A 'Y' IN REPLY TO ACTIVATE THE DELETE REST OPTION, OTHERWISE THE ORIGINAL SELECTIVE DELETE PROMPT WILL BE DISPLAYED.

Q = QUIT PROCESSING THIS UIC, PROCESSING WILL CONTINUE WITH THE NEXT UIC IF A WILDCARD GROUP OR MEMBER WAS SPECIFIED.

** NOTE **

WHEN PROCESSING WILDCARD GROUPS OR MEMBERS, THE 'G' OPTION ONLY APPLIES TO THE UIC IN WHICH IT WAS ENTERED. IT WILL NOT CARRY FORWARD TO THE NEXT UIC.

>> DELETE FILE /DE

THIS SWITCH CAUSES SRX TO AUTOMATICALLY DELETE ALL SELECTED FILES. WHEN USED, A '/LI' FORM DISPLAY WILL BE OUTPUT FOR EACH FILE DELETED. THIS OPTION IS NOT ALLOWED FOR USAGE WITH THE MFD.

>> PURGE /PU[:N]

THIS SWITCH CAUSES SRX TO DISPLAY ALL EXCEPT THE LATEST 'N' VERSIONS OF A FILE FOR SELECTIVE DELETE. THE FORMAT OF THE DISPLAY IS IDENTICAL TO THE '/SD' FORMAT. IF 'N' IS OMITTED, THE DEFAULT IS ONE, IN OTHER WORDS ALL EXCEPT THE LATEST VERSION WILL BE DISPLAYED. THE PURGE OPTION WORKS ON THE NUMBER OF COPIES RATHER THAN ON THE VERSION NUMBER.

>> WRITEBACK UFD /WB

THIS SWITCH CAUSES SRX TO WRITEBACK AN UPDATED VERSION OF THE DIRECTORY (ONE WHICH HAS BEEN COMPRESSED AND SEQUENCED) TO THE SOURCE DISK. THIS FEATURE REDUCES THE FILES-11 SYSTEM OVERHEAD INCURRED WHEN SCANNING A DIRECTORY FOR A FILE, SINCE ALL UNUSED DIRECTORY SLOTS HAVE BEEN MOVED TO THE END OF THE DIRECTORY AND THE END-OF-FILE POINTER NOW POINTS TO THE LAST BLOCK ACTUALLY USED.

COI-SSUF11001

** NOTE **

THE '/WB' OPTION WILL BE AUTOMATICALLY CANCELLED WHEN SOME TYPE OF FILE SELECTION IS ACTIVE OR WHEN A PROBLEM IS ENCOUNTERED WHILE READING THE DIRECTORY INTO MEMORY.

DURING A '/WB' OPERATION, THE DIRECTORY IS INACCESSIBLE TO THE REST OF THE SYSTEM UNTIL IT HAS BEEN SUCCESSFULLY WRITTEN BACK. THUS, THE USER SHOULD NOT PERFORM THIS OPERATION ON AN OPERATIONAL SYSTEM UNLESS IT IS FAIRLY QUIESCIENT.

COI-SSUF11001

CONSOLE OUTPUT MESSAGES

SRX WILL DISPLAY ONE OF THE FOLLOWING MESSAGES WHEN A PROBLEM IS DETECTED. THE PROBLEM MAY OR MAY NOT BE FATAL. IN THE CASE THAT IT IS NOT, SRX WILL CONTINUE NORMAL PROCESSING, OTHERWISE, THE CURRENT COMMAND LINE OPERATION WILL BE CLOSED OUT (FILES CLOSED, ETC.) AND PROCESSING WILL RESUME WITH THE NEXT INPUT COMMAND LINE.

<...SRX>- COMMAND LINE READ ERROR
<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE IS DISPLAYED WHEN A PROBLEM IS ENCOUNTERED WHILE READING A COMMAND LINE FROM THE CONSOLE OR AN INDIRECT COMMAND FILE. THIS IS A FATAL ERROR AND SRX WILL EXIT AFTER DISPLAYING THE CONSOLE MESSAGE.

<...SRX>- COMMAND LINE SYNTAX ERROR

THIS MESSAGE IS DISPLAYED WHEN A COMMAND LINE FORMAT ERROR IS DETECTED BY THE COMMAND STRING INTERPRETER (CSI) WHILE SYNTAX CHECKING THE INPUT COMMAND LINE. PROCESSING WILL CONTINUE WITH THE NEXT COMMAND LINE.

<...SRX>- COMMAND LINE SYNTAX ERROR - INPUT SPECIFICATION

THIS MESSAGE IS DISPLAYED WHEN A COMMAND LINE FORMAT ERROR IS DETECTED BY THE COMMAND STRING INTERPRETER (CSI) WHILE PARSING THE 'INFILE' FILE-SPECIFICATION. PROCESSING WILL CONTINUE WITH THE NEXT COMMAND LINE.

<...SRX>- COMMAND LINE SYNTAX ERROR - OUTPUT SPECIFICATION

THIS MESSAGE IS DISPLAYED WHEN A COMMAND LINE FORMAT ERROR IS DETECTED BY THE COMMAND STRING INTERPRETER (CSI) WHILE PARSING THE 'SORTFILE' FILE-SPECIFICATION. PROCESSING WILL CONTINUE WITH THE NEXT COMMAND LINE.

<...SRX>- COMMAND LINE SYNTAX ERROR - SORT SPECIFICATION

THIS MESSAGE IS DISPLAYED WHEN A COMMAND LINE FORMAT ERROR IS DETECTED BY THE COMMAND STRING INTERPRETER (CSI) WHILE PARSING THE 'SORTFILE' FILE-SPECIFICATION. PROCESSING WILL CONTINUE WITH THE NEXT COMMAND LINE.

<...SRX>- CONFLICT IN VERSION SELECTION SWITCHES

THIS MESSAGE IS DISPLAYED WHEN THE 'INFILE' FILE-SPECIFICATION CONTAINS MULTIPLE VERSION SELECTION SWITCHES (/OV, /NV, /HV OR /PU). PROCESSING WILL CONTINUE WITH THE NEXT COMMAND LINE.

<...SRX>- INVALID DATE FORMAT

THIS MESSAGE IS DISPLAYED WHEN AN INVALID DATE HAS BEEN INPUT AS A SELECTION PARAMETER, THE DAY MUST BE IN THE RANGE OF 1-31 AND THE MONTH MUST BE A VALID THREE CHARACTER CODE (IE: JAN, FEB, ETC.). THE YEAR MUST BE 70 ARE GREATER (1970 IS THE BASE YEAR FOR MONOTOMIC DATE COMPUTATION). PROCESSING CONTINUES WITH THE NEXT COMMAND LINE.

<...SRX>- '/WB' CANCELLED

THIS MESSAGE IS DISPLAYED WHEN THE USER HAS REQUESTED THE '/WB' OPTION AND HAS SPECIFIED SOME TYPE OF FILE SELECTION OR A READ PROBLEM WAS ENCOUNTERED WHILE LOAD THE DIRECTORY, PROCESSING CONTINUES AS IF THE USER HAD NOT INPUT '/WB'.

<...SRX>- NO SUCH FILE(S)

THIS MESSAGE WILL BE DISPLAYED WHEN THE USER HAS SOME TYPE OF FILE SELECTION ACTIVE AND NO FILES WERE FOUND WHICH MET THE SELECTION PARAMETERS. THIS IS AN AUDIT TRAIL MESSAGE.

<...SRX>- UFD TOO LARGE FOR MEMORY, DOING IN PIECES

THIS MESSAGE WILL BE DISPLAYED WHEN A UFD CANNOT BE LOADED IN ITS ENTIRETY FOR SINGLE-SHOT PROCESSING. IF THE USER REQUESTED THE '/WB' OPTION, IT WILL BE CANCELLED. SRX WILL PROCESS THE DIRECTORY IN SEGMENTS. THUS, ANY SWITCHES WHICH WOULD REQUIRE THE ENTIRE DIRECTORY BEING AVAILABLE (IE: /PU, /OV, /NV, ETC.) FOR PROPER SELECTION, WOULD, IN ALL PROBABILITY, GENERATE INCORRECT OUTPUT DISPLAY'S.

<...SRX>- '/SF' CANCELLED

THIS MESSAGE IS DISPLAYED WHEN THE USER HAS REQUESTED AN OUTPUT 'SORTFILE' (/SF) BE CREATED AND AN INPUT 'SORTFILE' (/LS) BE LISTED. SRX WILL CONTINUE PROCESSING OF THE LIST REQUEST ONLY.

<...SRX>- FCS PARSE ERROR - 'FILE-SPECIFICATION' '(PC)'

<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE IS DISPLAYED WHENEVER SRX IS NOTIFIED OF AN ERROR WHEN PARSING ONE OF THE COMMAND LINE FILE-SPECIFICATIONS. SRX WILL CONTINUE PROCESSING WITH THE NEXT COMMAND LINE.

<...SRX>- FILE OPEN ERROR - 'FILE-SPECIFICATION'

<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE IS DISPLAYED WHENEVER SRX ENCOUNTERS A PROBLEM OPENING AN INPUT/OUTPUT FILE FOR PROCESSING. SRX WILL CLOSE ALL OPEN FILES, DISCONTINUE PROCESSING THE CURRENT COMMAND LINE AND CONTINUE PROCESSING WITH THE NEXT COMMAND LINE.

COI-SSUF11001

<...SRX>- FILE READ ERROR - 'FILE-SPECIFICATION'
<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE IS DISPLAYED WHENEVER SRX ENCOUNTERS A PROBLEM READING ONE OF THE INPUT FILES (SORTFILE, UFD, ETC.). SRX WILL CLOSE ALL OPEN FILES, DISCONTINUE PROCESSING THE CURRENT COMMAND LINE AND CONTINUE PROCESSING WITH THE NEXT COMMAND LINE.

<...SRX>- FILE WRITE ERROR - 'FILE-SPECIFICATION'
<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE IS DISPLAYED WHENEVER SRX ENCOUNTERS A PROBLEM WRITING ONE OF THE OUTPUT FILES (SORTFILE, UFD, ETC.). SRX WILL CLOSE ALL OPEN FILES, DISCONTINUE PROCESSING THE CURRENT COMMAND LINE AND CONTINUE PROCESSING WITH THE NEXT COMMAND LINE.

<...SRX>- FILE HEADER ERROR - 'FILE-SPECIFICATION'
<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE IS DISPLAYED WHENEVER SRX ENCOUNTERS A PROBLEM READING THE FILE HEADER FOR A FILE. IF AN OUTPUT LIST FILE IS BEING CREATED, SRX WILL INCLUDE THE MESSAGE IN THE FILE PRECEDED BY A BLANK LINE AND FOLLOWED BY A BLANK LINE. SRX WILL CONTINUE PROCESSING WITH THE NEXT SELECTED FILE.

<...SRX>- FILE DELETE ERROR - 'FILE-SPECIFICATION'
<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE IS DISPLAYED WHENEVER SRX ENCOUNTERS A PROBLEM DELETING A FILE (/SD OR /DE). IF THE USER REQUIRES THAT THE FILE BE DELETED, IT WILL HAVE TO BE MANUALLY ACCOMPLISHED AFTER THE PROBLEM HAS BEEN CORRECTED. PROCESSING CONTINUES WITH THE NEXT SELECTED FILE.

<...SRX>- CREATED FILE - 'FILE-SPECIFICATION' <PAGE OR RECORD COUNT>
<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE IS DISPLAYED ANYTIME SRX CREATES AN OUTPUT FILE. THE FULL FILE-SPECIFICATION IS GENERATED ALONG WITH THE VOLUME OF DATA. IF IT A LIST FILE, THE NUMBER OF PAGES CONTAINED IN THE FILE WILL BE DISPLAYED, OTHERWISE IT WILL BE THE NUMBER OF RECORDS.

<...SRX>- ...SRT SPAWN REQUEST FAILED
<...SRX>- 'SYSTEM ERROR MESSAGE TEXT'

THIS MESSAGE WILL BE DISPLAYED IF SRX CANNOT SPAWN THE PDP-11 SORT TASK TO SORT THE OUTPUT SORTFILE. THIS NORMALLY OCCURS WHEN THE REQUIRED TASK IS NOT INSTALLED. THE PREVIOUS MESSAGE LINE CONTAINS THE COMMAND LINE WITH WHICH THE SORT TASK WAS

COI-SSUF11001

SPAWNED. AFTER CORRECTING THE PROBLEM, THE USER MAY MANUALLY
RE-EXECUTE THE SORT TASK WITH THE COMMAND LINE GIVEN.

** NOTE **

SRX CONVERTS THE QUEUE I/O AND DIRECTIVE STATUS WORD ERROR CODES FROM
A BINARY NUMBER TO AN INDEX FOR THE MESSAGE OUTPUT (MO) HANDLER SO THAT
AN ASCII TEXT MESSAGE WILL BE DISPLAYED IN PLACE OF THE NEGATIVE NUMBER.
IF SOME TYPE OF PROBLEM IS ENCOUNTERED DURING MO OPERATIONS, SRX WILL
CONVERT THE ERROR CODE INTERNALLY AND DISPLAY ONE OF THE FOLLOWING
LINES IN PLACE OF THE 'SYSTEM ERROR MESSAGE TEXT' LINE.

<...SRX>- DSW ERROR: (ERROR CODE)
<...SRX>- I/O ERROR: (ERROR CODE)
<...SRX>- UNK ERROR: (ERROR CODE)



