



**The**  
**- IAS -**  
**- DeVIAS -**  
**Newsletter**

**August 1984 Issue**

( )

( )

( )

From the Editor

We are now beginning our second year under the Decus paid-subscription service; but, it will be our first year as an independent newsletter. The level of subscription and the quality of submissions will determine whether the newsletter will continue. A target of 8 issues has been set for the next year and every effort will be directed at achieving that level. Co-operation is necessary, particularly, from Digital and the long-term IAS users. Participation, however is needed from all of you who have an interest in IAS.

Contributions

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

Please send all contributions to:

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

Attn: J. Drummond  
Mail Stop M5-D1

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.

Department of Radiation Therapy  
School of Medicine  
University of Pennsylvania  
Room 410  
133 South 36th Street  
Philadelphia, Pennsylvania 19104  
30 July 1984

Dear IAS Reader,

This issue, Number 22 I think, is the first issue of the DeVIAS Letter in the Fiscal Year 1985. The implications of this are (1) a new format and (2) being alone. I haven't seen the format and will be as surprised as you. Being alone means that our circulation will be decreased because the mass of RSX-11M readers will not get to read this as an extra - They'll have to pay separately to read about IAS. And you will have to pay separately to read about RSX. I'm not sure who conceived this scheme, but DECUS is bound to try it. So, for a year, at least, the DeVIAS Letter will be a separate publication again. I am glad in many ways. Most of the material contained herein is not of interest to the thousands of RSX readers who should not feel that they are paying for our newsletter. Our newsletter should stand and fall on our own subscriptions. When the interest wanes (subscriptions fall) we should quit.

I was talking to Ken Guralnik (E,G & G, Las Vegas) today on the phone. He's submitted a session for Anaheim that I am pushing:

#### My Favorite Macro

He's looking for you MACRO programmers to contribute a 2 to 5 minute presentation on your favorite macro. With examples of its use. You're shy, you say? Send him the code and explain it to him and he'll present it for you. We'll publish all of them in the DeVIAS Letter. Since MACRO-11 is spread about many operating systems, tell your friends; it is not IAS specific, nor is it meant to be neat tricks with IAS. It's neat tricks with MACRO! To show you the idea, I'll append my version of PUSH and POP to this letter. You can call Ken at 702-295-2921. Can't come to Anaheim? Please send your contribution to me so I can share it here and at Anaheim. Thanks.

I like MACRO-11 (and am thrashing with MACRO-32) and have encouraged the production of some more sessions at the symposia. But, for many of us, this is the medium that gets around.

At Anaheim, the IAS SIG is presenting two Pre Symposium Seminars of MACRO - one similar to the successful venture of Cincinnati but aimed at high level programmers who haven't used assembler before. Bob Agnew, who teaches Assembler at Beaver College; Bob Mack, who now works for the Government Systems Group at Digital and I are presenting this one. The second is MACRO-11: The Good Stuff. Mike Reilly and Allison Nylander, of the IAS Development Team at Digital, are presenting this gourmet offering. During the week Jim McGlinchey will present a two hour tutorial on Super-Mac, a structured programming approach to assembler programming. In fact, Jim has agreed to coordinate SIG efforts to promote MACRO-11. Thus, with the My Favorite Macro session, we are bring this important tool back to the national symposium.

A teaser from Bob Schuldt (INCO, Inc., Fairfax, VA) or was it Merton Crockett (I can't remember):

JSR R0,(PC)  
JSR R1,(PC)  
JSR R2,(PC)  
JSR R3,(PC)  
JSR R4,(PC)  
JSR R5,(PC)

What is accomplished? What is the state of the PSW after executing this program fragment?

For those of you with Rainbows - Poly-XFR works on IAS v3.2! I hear that the RSX-11M version can be patched to work with the new functionality in the terminal handler. Ken Pauwels (University of North Carolina, Chapel Hill) said that Polygon Associates have provided him with the patches directly.

After Cincinnati there are some changes in the Sig structure that I'd like to report. I have accepted the resignations of Bob Stodola (Institute for Cancer Research, Fox Chase, PA) and Tim Mahaney (Naval Air Propulsion Center, Trenton, NJ) from the steering committee. They were two of the five signatories to the document petitioning DECUS to grant a license to the Delaware Valley IAS LUG which became widely known as the DeVIAS LUG. They have both moved in job responsibility and away from IAS to the point they felt that they couldn't contribute at the same level any longer. I thank both of you for your long contributions to the IAS community.

There are, then two new faces on the steering committee: Mike Reilly (Digital, Maynard) and Mike Robitaille (Grumman-CTEC, Inc., McLean, VA). Mike Reilly is the now famous expert in the inner workings of IAS (and RSX and VMS and POS). He will deal with issues relating to the technical side of the SIG. Mike Robitaille is our new librarian. You want some program, you want to contribute a program? Talk to him.

Michael A. Robitaille  
Grumman-CTEC, Inc.  
6862 Elm Street  
McLean, VA 22101  
(703)-556-7400

Thus, we have:

Ron Fussell (AFIS, Washington, DC) - Symposia Coordinator  
John Drummond (Ontario Hydro, Toronto) - Newsletter Editor  
Ken Guralnik (E, G & G, Las Vegas) - WHIMS Coordinator  
Ray French (Boeing Aircraft, Seattle) - RSX-11M SIG Liason  
George Wells (General Electric, Louisville, KY) - Membership  
Mike Reilly (Digital, Maynard) - Technical Issues  
Mike Robitaille (Grumman-CTEC, McLean, VA) - Librarian  
Tim Leisman (Digital) - Digital-Sig Counterpart  
Bob Curley (U of P, Philadelphia) - Chairman

In addition Bob Schuldt (INCO, Inc., Fairfax, VA) and Skip Standfield (AFIS, Washington, DC) have agreed to help Mike Robitaille organize the library.

An unpleasant side effect of Cincinnati - Someone signed a room service slip for the IAS Suite and Ron Fussell and I are left to pay it - personally. Also the phone bill. Not charge to DECUS, personally. I'm not sure that I dare arrange an IAS Suite for Anaheim.

I heard a rumor, from a credible source, that Version 3.3 is being planned. And Version 4.0 is being planned for a year later. Remember to fill in your WHIMS forms and get them to Ken Guralnik. There was a huge spirit of cooperation at Cincinnati that was wonderful to watch. Our guys (and gals) at Digital are really interested in providing new and useful features for us.

Enough for this issue, read the rest of the good stuff that's been submitted,

Happy Summer Vacation,  
Bob Curley

```
.MACRO PUSH,REGLIS                ; <<< PUSH >>>
.IF NB,<REGLIS>                   ;
.IRPC REG,<REGLIS>                 ; Invocation: PUSH <012> or PUSH
MOV  %REG,-(SP)                   ; Push the indicated registers
onto
.ENDM                               ; the stack. If no register list
.IFF                               ; is used a PUSH <012345> is
.IRPC REG,<012345>                 ; assumed.
MOV  %REG,-(SP)                   ;
.ENDM                               ;
.ENDC                               ;
.ENDM                               ;
;
;-----
.MACRO POP,REGLIS                 ; <<< POP >>>
.IF NB,<REGLIS>                   ;
.IRPC REG,<REGLIS>                 ; Invocation: POP <210> or POP
MOV  (SP)+,%REG                   ; The order should be reversed
from
.ENDM                               ; the PUSH specification if
specific                           ; registers are named.
.IFF                               ; A POP without argument is
.IRPC REG,<543210>                 ; assumed
MOV  (SP)+,%REG                   ; to be POP <543210>
.ENDM                               ;
.ENDC                               ;
.ENDM                               ;
;
```

July 25, 1984

Robert Curley  
PO Box 322  
Flourtown, PA 19031-0322

Dear Bob:

As we discussed previously over DCS, the National LUG Organization will not be renewing the LUG License for DEVIAS. To reiterate, the reason is simply that LUGs serve local interests, not international. Since local interest is dead, IAS will be able to serve its followers as a SIG.

If in the future you plan to start a "generic" DECUS user group in the Philadelphia area, please do not hesitate to call me for assistance at 617-480-3289.

Thank you for your active participation in our Society.

Regards,



Anne K. Foley  
Local User Group Administrator  
DECUS U.S. Chapter

/gc  
enclosure

cc: Robert B. Robbins  
Randall S. Brown  
Joseph Ryan



**SOFTWARE PERFORMANCE REPORT**

FIELD NO.:	CORPORATE SPR NO.:
------------	--------------------

403537

✓ TO SET UP FOR PROPER ALIGNMENT, START AT MARK BELOW.

PAGE 1 OF 1

OPERATING SYSTEM IAS	VERSION 3.1	SYSTEM PROGRAM OR DOCUMENT TITLE IAS DEMO (MCR function)	VERSION OR DOCUMENT PART NO. 3.1	DATE 07 June 84
NAME: Air Force Intelligence/IND FIRM: Bolling AFB, Bldg 5681 Washington, DC 20332		DEC OFFICE Honolulu POC-Layton Fuji		DO YOU HAVE SOURCES? YES <input type="checkbox"/> NO <input checked="" type="checkbox"/>
ADDRESS: Attn: Lt. Ron Fussell		REPORT TYPE/PRIORITY		
CUST. NO.: 8305 4234J		<input checked="" type="checkbox"/> PROBLEM/ERROR <input type="checkbox"/> SUGGESTED ENHANCEMENT <input type="checkbox"/> OTHER		
SUBMITTED BY: Michael Scott Ward		PHONE: (808)-477-6256		
ATTACHMENTS MAG TAPE <input type="checkbox"/> FLOPPY DISKS <input type="checkbox"/> LISTING <input type="checkbox"/> DECTAPE <input type="checkbox"/>		CAN THE PROBLEM BE REPRODUCED AT WILL? YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>		
OTHER: NON-APPLICABLE		COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? YES <input type="checkbox"/> NO <input checked="" type="checkbox"/> PLEASE EXPLAIN IN PROVIDED SPACE BELOW.		
CPU TYPE PDP 11/70	SERIAL NO. BT17948	MEMORY SIZE 1024 DEC WD	DISTRIBUTION MEDIUM 9 Tr Magtape	SYSTEM DEVICE BR 1538D
				DO NOT PUBLISH <input type="checkbox"/>

Problem: Problems were detected with the IAS 3.1 display memory program (DEMO) when we increased our terminals beyond 77(octal)/63.(decimal). This resulted in three-digit terminal unit numbers being displayed appropriately with the active task, which in turn corrupted the "NODES=" and "LARGEST HOLE=" headers on the same line. Additionally, when a different active task with only a two-digit terminal unit number is displayed immediately after, the third digit from the previous display image is not cleared from the screen. The example below will reproduce the detected errors on a DEC display terminal.

Create the program below, assemble and taskbuild for testing.

```

;      (1,1)XXX.MAC      // Create a second test program, by renaming the program
                        // to the left as follows.
START: CLR RO           //
        INC RO          //
        BNE START      //
        .END START     //
    
```

Next perform the following operations to create a 3-digit test device, PI is used in this case, but any unused device may be created in the same manner.

```

MCR DEV PI/PUD      (output = PIO 146530)
MCR OPE 146530/KNL (CR)
    00146530 044520/ (CR) (output)
    00146532 000000/101$ (ESC) (output)
MCR INS XXX
MCR RUN XXX/TI=PI101/PRI=45
MCR INS YYY
MCR RUN YYY/PRI=50
MCR DEM (CR) (must be issued from a DEC display terminal)
              (monitor screen until problems are seen)
    
```

CY TO: Inco, Inc.  
IPAC, Box 38 (sm-3)  
Camp Smith, HI 96861 Attn: Michael Scott Ward

ALL SUBMISSIONS BECOME THE PROPERTY OF DIGITAL EQUIPMENT CORPORATION.

MNT. GRP.	XFER GRP.	PL	PRB. TYPE
DATE TO MAINTAINER	XFER DATE	LOGGED ON	
DATE RECEIVED FROM MAINTAINER	DATE ANSWERED	LOGGED OFF	





**SOFTWARE PERFORMANCE REPORT**

FIELD NO.	CORPORATE SPR NO.
-----------	-------------------

817287

✓ TO SET UP FOR PROPER ALIGNMENT, START AT MARK BELOW.

PAGE \_\_\_\_ OF \_\_\_\_

OPERATING SYSTEM IAS	VERSION 3.1	SYSTEM PROGRAM OR DOCUMENT TITLE IAS ACT (utility)	VERSION OR DOCUMENT PART NO. 3.1	DATE 11 Jun84
NAME: Air Force Intelligence/IND FIRM: Bolling AFB, Bldg 5681 Washington DC 20332  ADDRESS:  CUST. NO.: 8305 4234J		DEC OFFICE AND CONTACT PERSON POC-Layton Fuji Honolulu		DO YOU HAVE SOURCES? YES <input type="checkbox"/> NO <input checked="" type="checkbox"/>
SUBMITTED BY: Michael Scott Ward		PHONE: (808) 447-6256		CAN THE PROBLEM BE REPRODUCED AT WILL? YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>
ATTACHMENTS MAG TAPE <input type="checkbox"/> FLOPPY DISKS <input type="checkbox"/> LISTING <input checked="" type="checkbox"/> DECTAPE <input type="checkbox"/>		OTHER: COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? YES <input type="checkbox"/> NO <input checked="" type="checkbox"/> PLEASE EXPLAIN IN PROVIDED SPACE BELOW.		
CPU TYPE PDP 11/70	SERIAL NO. BTI7948	MEMORY SIZE 1024 DECWD	DISTRIBUTION MEDIUM 9 Tr magtape	SYSTEM DEVICE BR 1538D
DO NOT PUBLISH <input type="checkbox"/>				

**PROBLEM:** The IAS 3.1 active task MCR function program (ACT) displays an invalid terminal number when the TI number exceeds 77 (octal) or 63 (decimal). When a three-digit terminal unit number is displayed by ACT, it truncates the first number from being displayed. The attached listing contains the procedures to reproduce the problem.

**DIAGNOSIS:** We are experiencing several problems with IAS utilities which reference or display terminal numbers. Gathered from our extensive research it was determined that ACT does not allow terminal unit numbers above 63., due to limitations set. It is inconceivable why DEC restricted many of their utilities to handle only terminal unit numbers below 64., this is somewhat abusive to structured programming techniques!!

Cy to: INCO, Inc.  
IPAC, Box 38 (sm-3)  
Camp Smith, HI 96861 Attn: Michael Scott Ward

ALL SUBMISSIONS BECOME THE PROPERTY OF DIGITAL EQUIPMENT CORPORATION.

SHORT NAME	MNT. CAT.	MNT. GRP.	XFER GRP.
DATE RECEIVED (MAIL)	DATE TO MAINTAINER	DATE RECEIVED (ASG)	DATE ANSWERED

The examples given below will reproduce the detected errors.

The first example shows the task file needed for testing and the appropriate procedures for setting up the test device. PI was used in this case, but the null device or any other which is not currently in use may be set up for testing in the same manner.

```
;  
; E1,13XXX.MAC (Assemble & Taskbuild)  
;  
;  
START:  
    CLF      RO  
    INC      RO  
    BNE      START  
    .END     START  
MCR>DEV PI/PUD  
    PIO 146530  
MCR>OPE 146530/KNL  
00146530 044520/  
00146532 000000/101# (esc)  
  
MCR>INS E1,13XXX  
MCR>RUN XXX/TI=PI101/PRI=45  
MCR>ACT XXX/TI=PI101  
ACT -- INVALID DEVICE  
MCR>
```

Initially act produces an error message as shown, this was zapped to reflect a valid device as shown below.

```
ZAP  
ZAP>E11,13ACT  
_3:2500+356/ (cr)  
003:003056/ 000004  
_5 (cr)  
_X (cr)  
MCR>ACT XXX/TI=PI101  
XXX RUN TS PI01  
MCR>
```

Although the device is now valid, the unit number is truncated.

PLEASE READ ATTACHED INSTRUCTIONS

PLEASE TYPE



SOFTWARE PERFORMANCE REPORT

FIELD NO.: CORPORATE SPR NO.:

071339

TO SET UP FOR PROPER ALIGNMENT, START AT MARK BELOW.

PAGE 1 OF 2

Form with fields: OPERATING SYSTEM (IAS), VERSION (3.1), SYSTEM PROGRAM OR DOCUMENT TITLE (IAS PIP (UTILITY)), VERSION OR DOCUMENT PART NO. (3.1), DATE (17 May 84), NAME (Air Force Intelligence/IND), FIRM (Bolling AFB, Bldg 5681), ADDRESS (Washington, DC 20332), Attn: Ron Fussell, CUST. NO.: 8305 4234J, SUBMITTED BY: Michael Scott Ward, PHONE: (808) 477-6256, CAN THE PROBLEM BE REPRODUCED AT WILL? YES [X] NO [ ], ATTACHMENTS (MAG TAPE, FLOPPY DISKS, LISTING, DECTAPE), OTHER: Non Applicable, CPU TYPE (PDP 11/70), SERIAL NO. (BT 7948), MEMORY SIZE (1024 DEC WD), DISTRIBUTION MEDIUM (9 Tr Magtape), SYSTEM DEVICE (BR 1538D), DO NOT PUBLISH [ ]

PROBLEM: The IAS 3.1 peripheral interchange utility program (PIP) produces an invalid task exit status when attempting to append a file via the /ap switch. When a failure occurs during a PIP append operation, a successful exit status is returned as shown in the example below. Use the IND (indirect command file processor) to execute the following tests.

@piptst.cmd (hit escape)

Create the following command files in [1,1] to reproduce the error and to validate other IAS 3.1 PIP operations.

```
.; [1,1]PIPTST.CMD;1
.DISABLE LOWERCASE
SET /UIC=[1,1]
.OPEN FILE 2
.DATA [1,1]FILE2.DAT;1
.CLOSE
.IFNINS ...PIP INS [11,1]PIP
PIP FILE1.DAT=FILE2.DAT/AP
.IF EXSTAT = SUCCES .GOTO 1
; NON-SUCCESSFUL
.GOTO 2
.1:
; SUCCESSFUL
.2:
.ENABLE LOWERCASE
```

```
.; [1,1]PIPTST1.CMD;1
.DISABLE LOWERCASE
SET /UIC=[1,1]
.IFNINS ...PIP INS [11,1]PIP
; VALIDATION OF PIP 3.1 EXIT WITH NO
: SWITCHS
PIP XXX.DAT=YYY.DAT
.IF EXSTAT = SUCCES .GOTO 1
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
;NON-SUCCESSFUL-PIP NON-SWITCH TEST.
.GOTO 2
.1:
; SUCCESSFUL-PIP NO-SWITCH TEST.
.2:
; TO PROCESS THE NEXT TEST PROCEDURE
; TYPE CNTRL/C, RES ...AT. (hit escape)
.PAUSE
; VALIDATION OF PIP 3.1 EXIT STATUS
; WITH /RE RENAME SWITCH.
.3:
PIP XXX.DAT;1=YYY.DAT;1/RE
(cont on 2)
```

CY TO:

INCO, INC. IPAC, Box 38 (SM-3) Camp Smith, HI 96861 Attn: Michael Scott Ward

Ontario Hydro 700 University Ave. Toronto, Canada Canada, MSG 1X6 Attn: Mr John W. Drummond Mail Stop - M2E10

ALL SUBMISSIONS BECOME THE PROPERTY OF DIGITAL EQUIPMENT CORPORATION.

Table with 4 columns: SHORT NAME, MNT. DAT., MNT. GPP, XFER GPP; DATE RECEIVED (MAIL), DATE TO MAINTAINER, XFER DATE; DATE RECEIVED (ASG), DATE RECEIVED FROM MAINTAINER, DATE ANSWERED

SPR  
SOFTWARE PROBLEM REPORT

(CONTINUED FROM PAGE 1)

[1,1]PIPTST1.CMD;1

```
.IF <EXSTAT> = <SUCCES> .GOTO 4  
; NON-SUCCESSFUL-PIP/RE TEST  
.OPEN YYY  
.DATA [1,1]YYY.DAT;1  
.CLOSE  
.GOTO 3  
.4:  
; SUCCESSFUL-PIP/RE TEST  
.ENABLE LOWERCASE
```

DIAGNOSIS:

My conclusion from the above samples. When using the append switch with PIP the appropriate exit status flags are not conditionally being set.

CURE:

Unknown, do not possess source.

Please reply upon receipt & determination.

Thank you,

*Michael Scott Ward*

Michael Scott Ward (SM-3)

## Two Terminal Handlers in IAS

John Roman  
McDonnell Douglas Corporation  
5775 Campus Parkway  
Hazelwood, Missouri 63042  
(314) 234-0984

### 1.0 Introduction

We have an IAS V3.0 system on 11/44s using the dumb terminal handler (TT01) as the console terminal handler. Printers and block mode intelligent CRTs are attached to DZ-11s using custom handlers. We wanted to add dumb CRTs (or micros to emulate dumb CRTs) on DZ-11s, yet keep the TT01 for the console, as we had modified it in a manner which would be difficult for the smart terminal handler (TT). Thus we wanted to run our system with two terminal handlers, TT01 for the console and the TT for CRTs off the DZs.

This article describes how we went about this. Looking back on the process, it seems almost trivial, but we torched a few packs on the way.

### 2.0 Setting Up the Second Terminal Handler

The effort was devoted to assembling and building the smart terminal handler as the second terminal handler in the system. The work involved changes to TT and sysgen changes.

#### 2.1 Changes to the Smart Terminal Handler

We made several changes to the terminal handler modules. The first was to CONFIG.MAC, where we edited the INTF macros to reflect our system. Essentially, we commented out the DL line and modified DZ lines to start at interface 0 and indicate the vector and CSRs for our DZs. Then the TERM macros were modified to indicate the terminals we had connected to the DZs.

The second change was to PARAMS.MAC, where we indicated there were no DLs in the system, and the proper number of lines and support for dial-up lines, etc.

The third change was to DATA.MAC, where we edited the TTNAME equate to indicate the name under which the handler was to declare residency.

We also changed the task build command file to indicate the new task name.

The fourth change was to IDLE.MAC, where we added support for control-X with different task names than TTYNxx, where xx is the line number. The change we made has the handler use the TTNAME equate to determine its name and request the task zzYNxx, where zz are the two characters in TTNAME.

The code we added is as follows:

```
MOV     R2,-(SP)           ;Push affected registers
MOV     R3,-(SP)           ;
MOV     #TTNAME,R2        ;Get the task name
MOVB    R2,R3             ;Get the low byte
BICB    #100,R3           ;Remove ascii bias to get Rad50
MUL     #50,R3            ;and shift up to accumulate
SWAB    R2               ;Get the high byte
BIC     #177500,R2        ;Clear off the unnecessary bits
ADD     R2,R3             ;Add to R50 accumulator
MUL     #50,R3            ;Shift up
ADD     #31,R3            ;Add a Rad50 'Y' after remove bias
MOV     R3,CXDPB+R.QSTN   ;Put into request DPB
MOV     (SP)+,R3          ;And restore affected registers
MOV     (SP)+,R2          ;
```

This code should be added to the routine TACNTX (Task action for control-X) in the IDLE module. It should be placed after the check for slaved terminal. When this change is made, install tasks as desired with the proper name and they will be requested when control-X is pressed.

## 2.2 Sysgen Changes

The main changes required are to add PUDS in Phase 1 for the terminals on the DZ lines. In Phase 2, we have the second terminal handler installed and loaded before the system is saved.

## 3.0 System Operation

The system runs entirely as expected, with all MCR functions running properly (PIP, EDT, etc.). The only problem is that both terminal handlers must use the same time-out value, since it is a global value, .MCRT0, in SCOMM. We would prefer to have different time-outs for each handler, with the console terminal handler much shorter than for

the other.

#### 4.0 Conclusions

This attempt was entirely an empirical study as opposed to a theoretical study, in that we went ahead and tried it before we researched the system to determine if it would work. As it indicates, there is no reason that multiple copies of the terminal handler cannot be utilized. The procedure is straightforward, and the only necessary parts are the changes to CONFIG and PARAMS, the task name change, and the sysgen changes.

## EXECUTIVE ODT (EXDT)

### PART I: Activating EXDT and a Command Summary

by: Alison Nylander - IAS Software Developer; DEC

For those of you who missed my DECUS presentation on Debugging Device Handlers using Executive ODT, I'm running a two part article here in the newsletter to share this information with you. The first part (which is what you are reading at the moment) will cover how to get EXDT (Executive ODT) into your system and all of the commands available to you. The second part, in the next newsletter, will have examples of using EXDT and helpful (I hope) hints on slick ways to use it in debugging various and sundry programs.

So we begin.....

### THE MYSTERY OF ^C AND ^X

The first thing to do is to include EXDT in your EXEC. To do this edit LB:[11,15]EXEC.ODL. Change the reference of 'NODT' to 'ODT'. The following is an excerpt of EXEC.ODL to show what this looks like.

```

;
; [ 1 1 , 1 5 ] E X E C . O D L
;
; OVERLAY DESCRIPTOR FILE FOR IAS EXECUTIVE
;
;
;
; ODT OPTION
;
; CHANGE 'NODT' TO 'ODT' TO INCLUDE EXECUTIVE ODT IN THE SYSTEM
;
ODT:  .FCTR  [11,15]EXEC/LB:NODT
;
;
;
;
; .END
```

- o Remove the 'N' from 'LB:NODT' (in LB:[11,15]EXEC.ODL)
- o Rebuild the Executive (TKB @LB:[11,15]EXECBLD)
- o Sysgen

If you were wondering what the module NODT and ODT were, here is a brief description of the two modules.

NODT - is a dummy module built into the Exec when EXDT is

not desired. This is the way your system is when you receive it on the tape.



ODT - is the module for the Executive debugger known as EXDT.  
This is the module you may want in your Executive.

Both modules live in LB:[311,15], both objects live in  
LB:[11,15]EXEC.OLB. (You will only have [311,15]  
if you have a source kit.)

Now that you have rebuilt the EXEC, you must do a SYSGEN to get the new  
improved version running on your system.

During Sysgen Phase 2 you will receive the following prompt:

```
IAS V03.2 (EXDT) - Type ^C to start  
*
```

You MUST type ^X here to be able to use EXDT.

After you have SAVED the system you will see the prompt again. THIS TIME you  
may type either ^X or ^C.

^C indicates that you will not be using EXDT until you reboot the system.  
^C tells EXDT that upon entry it is to JMP to it's termination point. It  
will NOT process any breakpoints. ^C says you will just run normally.

^X indicates that you may be using EXDT while the system is up. ^X tells  
EXDT to process breakpoints and to do all of the other wonderful things that  
EXDT does for you.

(and for those hackers out there....)

Here's briefly how it works:

When you BOOT the system. The boot code jumps to EM03 in it's Power Recovery  
Recognition Routine where it JMPs to EXDT. (So if you want to test the  
power fail code in your handler you've got a chance to get at it by setting a  
breakpoint at this time.) EXDT returns via the RTT or RTI instruction  
(depending on the CPU you're running on.)

Now EXDT is in your system. The following are the commands for EXDT and  
some brief examples. I have grouped the commands into general categories.

## ACTION COMMANDS

**BREAKPOINTS (r;B)** - Type r;B to set a breakpoint where r is the address of the location for the breakpoint. EXDT replaces the contents of the location with a BPT trap. When the breakpoint occurs the contents of the instruction are restored. There are eight breakpoints (0-7). Typing r;B will set the next available breakpoint. Specify a selected breakpoint by typing r;nB where n is a number 0-7.

```
*1020;B
*1030;B
*1040;B
*1032;1B
```

;B clears all of the breakpoints. ;nB clears only the nth breakpoint.

```
*;B
*;2B
```

\$B prints the first word of the breakpoint table. (See internal registers for more detail.)

**RUNNING THE PROGRAM (;G and r;P)** - Type either ;G or ;P to begin program execution. Execution will continue until either the program terminates or a breakpoint is encountered.

Type ;P after the first breakpoint has been encountered.  
Type r;P to loop r times through a breakpoint without stopping.

```
*1026;3B
*4;P
B3;001026 ; halts on the fourth time thru
```

**SINGLE-INSTRUCTION MODE (;nS)** - Type ;nS to execute n instructions at a time, where n is any digit, usually 1. Breakpoints are disabled.

Type n;P to allow the program to run for the next n instructions. If n is missing it is assumed to be 1. Trapping instructions can affect the proceed repeat count.

Type ;S to disable single instruction mode.

```
B3;001010
*;1S
*;P
001012
*
.
.
.
*;S
```

RELOCATION REGISTER COMMANDS (r;nR) - Type r;nR to preset the relocation biases of the relocatable modules to be debugged, where r is a relocatable expression and n is an integer 0-7. (n defaults to 0)

```
*1344/10046
*1000;1R
*1,344/10046
```

```
*20446;B
*20000;3R
*;B
*3,446;B
```

If multiple relocation registers are defined, EXDT uses the relocation register closest to the address in question.

RELOCATION CALCULATORS (n! and nR) - Type n! to calculate the relocatable address of an opened location, where n is the relocation register (n defaults to the closest register (but less than or equal to.))

```
*2000;3R
*2500/10046 !=3,000500
```

Type nR to calculate the difference between the contents of the opened location and a relocation register, where n is the relocation register.

```
*7000;1R
*1,500/011032 1R=1,002032
```

The following example is using both relocation calculators at once:

```
*1000;1R
*2000;2R
*3000/006410 1!=1,002000 2!=2,001000 1R=1,005410 2R=2,00410
```

#### ACCESSING GENERAL REGISTERS 0-7

OPEN REGISTER (\$n) - Type \$n/ where n is the number (0-7) of the register to be opened. <LE>, <CR>, ^, or @ can be used when a register is opened.

#### ACCESSING INTERNAL REGISTERS

STATUS REGISTER (\$S) - Type \$S/ to see the contents of the program's status register. The status register contains the condition codes of the most recent operational results and the interrupt priority level of the program.

BREAKPOINT TABLE (\$B) - Type \$B/ to see the contents of the first word in the breakpoint table. The contents of this word is breakpoint 0. The next seven(7) locations contain the addresses of the other breakpoints.

MASK LOCATION (\$M) - Type \$M/ to see the contents of the mask location for a bit pattern search. The mask is used in WORD search and Effective Address search. Bits set to '1' are used during the search. (More on this under 'Search'.)

CONSTANT REGISTER (\$C) - Type \$C/ to see the contents of the constant register. The constant register is used for converting relocatable addresses into their values after relocation. (More on this later.)

RELOCATION REGISTER (\$R) - Type \$R/ to see the contents of the first word of the relocation register table. This is relocation register 0. The next seven (7) locations contain the addresses of the other relocation values.

## OPENING, CHANGING, AND CLOSING LOCATIONS

SLASH (/) - Type the address followed by a slash. For example:

```
*1000/012746
```

where 1000 is the address and 012746 is the contents. If you do not want to change the contents of the address then type <CR> or <LF>. If you want to change the contents of the address type the new value then type <CR> or <LF>.

```
*1000/012746 12345 <CR>
```

BACKSLASH (\) - Type the address of the byte followed by a backslash.

```
*1001\101 =A
```

where 1001 is the address of the byte and 101 is the contents. Changing the contents is the same as with a SLASH.

LINE FEED KEY (<LF>) - Type <LF> after an open address and it closes the address and opens the next sequential location.

```
*1000/2340 <LF>  
001002/12740
```

If a byte was open, the <LF> opens the next byte.

UP-ARROW (^) - Type (^) and the previous location is opened.

```
*1002/12740 ^  
001000/2340
```

UNDERSCORE OR BACK-ARROW (\_) - Type (\_) or back-arrow to an open word EXDT interprets the contents of the word as an address indexed by the program counter (PC) and opens the addressed location.

```
*1006/000006 _  
001016 / 0000405
```

OPEN ADDRESSED LOCATION (@) - Type (@) to open the contents of the addressed location.

```
*1004/12456 @  
012456 / 000500
```

The contents of the address may be modified before being opened.

```
*1004/12456 1044 @  
001044/002004
```

RELATIVE BRANCH OFFSET (>) - Type (>) to close a location, it then uses the low-order byte as a relative branch offset to the next word to be opened.

```
*1032/000407 301>
000636 / 000010
```

Note that 301 is a negative offset (-77). EXDI doubles the offset before it adds it to the PC; (1034+(-176)=636).

RETURN TO PREVIOUS SEQUENCE (<) - Type (<) to close a location and then return to where it was before the () or (@) or (>) was typed.

```
*1032/000407 301>
000636 /000010 <
001034 /001040 @
001040 /000405 \005 = <
001035 \ 002 =? <
001036 \ 004 =?
```

## MORE USEFUL COMMANDS

RADIX-50 MODE (X) - Once you have a location open, type X and EXDT converts the contents of the location into it's three character ASCII equivalent. You can then either close the location OR type a new three character ASCII value. (Valid characters are '.', '\$', Space, 0 thru 9, A thru Z)

```
*1000/042431 X=KBI CBA <CR>
*1000/011421 X=CBA
```

WORD SEARCH (r;W) - Before initiating a word search, you must specify the mask and search limits. \$M/ opens the mask register. Set the bits that are to be examined. The next two sequential locations are the lower and upper limits of the search. Set the contents of these registers.

```
*$M/000000 177400<LF>
r,nnnnnn/000000 1000<LF>
r,nnnnnn/000000 1040<CR>
```

Next set the search object and the initiating command, r;W where r is the search object, ;W is the command. When EXDT finds a match (each bit in the search object is set to 1 in the word EXDT searches over the mask range), the matching word prints.

```
*400;W
001010/000770
001034/000404
*
```

Typing CTRL/U during a search printout terminates the search.

EFFECTIVE ADDRESS SEARCH (r;E) - Set the lower and upper limits for the search. (See the example in word search above.) Type the command r;E (where r is the effective address) to initiate the search.

Words that are an absolute address (argument r itself), a relative address offset, or a relative branch to the effective address print after the addresses. For example:

```
*$M/177400 <LF>
r,nnnnnn/001000 1010 <LF>
r,nnnnnn/001040 1060 <CR>

*1034;E
001016/001006 ;relative branch
001054/002767 ;relative branch
*1020;E
001020/177774 ;relative address offset
001030/001020 ;absolute address
```

Typing CTRL/U during a search printout terminates the search.

CONSTANT REGISTER (r;C) - Type r;C to evaluate the relocatable expression to its six-digit octal value, print it and store it in the constant register. r can be a relocatable address, or a number to be changed many places in the program. For example:

```
*-4432;C=173346
*6632/062701 C <CR>
*6632/173346
*1000;1R
*1,4272;C=005272
```

MEMORY BLOCK INITIALIZATION (;F and ;I) - Use the constant register, the lower and upper limits from search and the ;F and ;I commands to fill successive memory locations. ;F will fill successive words within the limits set with the contents of the constant register. ;I will fill successive bytes.

```
*$M/000000 <LF>
r,nnnnnn/000000 1,0 <LF>
r,nnnnnn/000000 2,-2 <CR>
*0;C=000000
*;F
*$M/000000 <LF>
r,nnnnnn/007000 2,0 <LF>
r,nnnnnn/007776 3,-1 <CR>
*40;C=000040
*;I
```

CALCULATING OFFSETS (r;D) - Type r;D to cause EXDT to print the 16-bit and 8-bit offsets from the currently open location to address r. For example:

```
*346/000034 414;D 000044 022 22 <CR>
*/000022
*1034/103421 1034;D 177776 377 \021 =? 377 <CR>
*/103777
```

The 8-bit offset only prints if it is in the range -128. to 127. and the 16-bit offset is even.

ASCII INPUT and OUTPUT (r;nA) - Type r;nA to inspect and change ASCII text. r represents a relocatable expression and n is the character count (defaults to 1). EXDT prints n characters starting at location r and followed by a <CR><LF> combination.

To change the contents of the words, type n characters of text after the <CR><LF>. Either type exactly n characters or fewer than n characters (terminated with a CTRL/U.)



The best way to become familiar with these commands is to take an evening and sit down at your console and experiment with them. A quick and easy way to get started (even though this should be in the next article) is to hard code a BPT instructions into the interrupt service routine of your favorite device handler (not the one driving the system disk). Then bring your machine up with ^X and run your handler. When the BPT is encountered you will find yourself (your machine actually) in EXDT. Try different commands and see what happens. The last page of this article is a little picture of memory at system startup time. I've included it to give you a bit of orientation to the inside of the system. You may find it useful when you're wandering around in there in EXDT. So have fun. Until the next newsletter.....

MEMORY AT SYSTEM STARTUP

APR 7	I/O PAGE	160000
APR 6	SCOMM .	140000
APR 5	SCOMM .	120000
APR 4	SCOMM .	100000
APR 3	SCRATCH	60000
APR 2	EXEC OVERLAYS	40000
APR 1	EXEC ROOT .	20000
APR 0	EXEC ROOT . (EXDT lives in here)	VIRTUAL 0
PHYSICAL 0	BOOT CODE	

June 16, 1984  
Frank R. Borger  
Instrumentation Division Head  
Department of Medical Physics

Robert F Curley  
Department of Radiation Therapy  
University of Pennsylvania  
Room 410  
133 South 36th Street  
Philadelphia, Pennsylvania 19104

Dear Bob:

One of the items on the IAS WHIMS list at the spring DECUS was the ability to change ones default device (if one is not running full-blown timesharing with PDS.)

The following patch to the operating system and new MCR utility achieve that result. they consist of two programs:

1. DEVPAT is run once at startup. It overwrites the SCOM routine ..FDEV to look for a default device pud in U.LBH of the tasks TI. NOTE that this patch only works for IAS version 3.1. If one wishes to adopt this to other versions, one must know the address of ..FDEV and do a disassembly to make sure ..FDEV hasn't changed. (As I remember, ..FDEV itself didn't change between 3.0 and 3.1, but don't hold me to it.)

DEVPAT is also an example of how we patch IAS. If anything goes wrong, we still boot with a virgin IAS system, and can insure our patches aren't screwing things up by just not executing the PATCH program.

2. SDV just scans the PUD for a match to the device specifier entered by the user. If a match is found, it puts that PUD address in U.LBH of the users TI pud. (Using the low maximum logical block number for a terminal device seems like an ok theft of a pud word, so far we havn't seen anybody else use it.)

Hope this will be of use to other IAS users out there.

Sincerely Yours,



Frank R. Borger  
FRB/pdp

```

;
; PROGRAM DEVPAT.MAC
; TO OVERWRITE CODE IN SCOMM, ROUTINE $$FDEV TO ENABLE
; INDIVIDUAL REDIRECTION OF THE SY DEVICE FOR REALTIME
; AS WELL AS TIME-SHARING TASKS. THE DEFAULT DEVICE
; INFORMATION FOR NON-TCP CONTROLLED TASKS IS IN THE
; PUD FOR THE TI AT OFFSET U.LBH. IT IS THE PUD ADDRESS
; OF THE DEFAULT DEVICE.
;
; THE TI PUD WILL HAVE TO BE FILLED IN FOR THE TIME
; BEING BY A SEPARATE TASK (SDV), POTENTIALLY ALLOWING
; DIFFERENT DEFAULT DEVICES FOR TCP AND NON-TCP TASKS.
; THIS TASK SHOULD BE RUN AT HIGH PRIORITY DURING INITIAL
; SYSTEM STARTUP TO PATCH EXEC. IF CONTEXT SWITCHING IS
; THOUGHT TO BE A PROBLEM, CODE SHOULD BE MODIFIED TO
; INHIBIT TASK SWITCHING.
;
; ONE MACRO NEEDED:
;
;         MCALL  EXIT$S
;
START:
MOV     #..FDEV+10,R0    ;GET START ADDRESS TO PATCH
MOV     #PATCH,R1      ;GET ADDRESS OF PATCH CODE
MOV     #14,R2          ;AND # OF WORDS FOR PATCH
1$:
MOV     (R1)+,(R0)+     ;HEAD 'EM UP AND
SOB     R2,1$           ;MOVE 'EM OUT
EXIT$S
;
.PAGE
; ++018
; ..FDEV IS CALLED TO SCAN THE PUD FOR A DEVICE NAME
;
; ON INPUT:      R0=TASK ATL
;                R4=DEVICE NAME IN ASCII
;                R5=UNIT NUMBER, (OCTAL)
;
; ON OUTPUT     R4 CLEARED
;                R5 = PUD ADDRESS (1 IF ITS TI)
;                C-BIT SET FOR ERROR, CLEARED FOR SUCCESS'
;
;
;
; TO ENABLE DEBUGGING, THE PC IS SET TO MATCH WHAT THE PC
; IS FOR ..FDEV FROM A DOB LISTING. THIS WILL HAVE NO EFFECT ON
; THE PATCH, SINCE IT ONLY DOES RELATIVE (BRANCHING) ADDRESSING
;
; =. +3276-26          ;PC NOW EQUAL TO SCOMM ADDRESS OF ..FDEV
;                    ;(SUBTRACT 26 TO ACCOUNT FOR ABOVE CODE)
;
; THIS IS THE CODE FOR ..FDEV
;
;
MOV     R5,-(SP)        ;SAVE UNIT #
MOV     R4,-(SP)        ;SAVE NAME
TST     R0              ;IS THERE AN ATL ADDRESS?
BEQ     20$             ;BR IF NOT

```

```

; THE FOLLOWING CODE IS OVERWRITTEN BY THE PATCH INSTRUCTION
; BIT      #AF. IA, A. TF(R0) ; TIMESHARING TASK?
; BEQ     10$                ; BR IF NOT
; TST     A. JN(R0)         ; TCP TASK?
; BEQ     10$                ; BR IF NOT
; CMP     (SP), # "SY"      ; IS THIS SY TO BE REDIRECTED ?
; BNE     10$                ; BR IF NOT
; MOV     A. TI(R0), R4     ; GET TI PUD
;
; THIS IS THE PATCH ITSELF
; START RELATIVE ADDRESS = (.. FDEV+10)
; PATCH=.                   ; DEFINE START OF PATCH CODE
MOV      A. TI(R0), R4     ; GET TI PUD POINTER
CMP      (SP), # "SY"     ; LOOKING FOR "SY" ?
BNE     10$                ; BR IF NOT, GET ACTUAL DEVICE
TST     A. JN(R0)         ; A TCP TASK ?
NOP     ; WAS BNE          NO-OP FOR MULTI-USER SYSTEM
;                               SO PATCH IS RIGHT # OF WORDS
MOV      U. LBH(R4), R5   ; GET DEFAULT DEVICE PUD ADDRESS
BNE     40$                ; BRANCH IF DEF PUD ADDRESS THERE
BR      10$                ; OTHERWISE LOOK FOR SY
;
; THE REST OF THIS CODE IS NOT OVERWRITTEN, BUT IS HERE SO
; BRANCH INSTRUCTION GET THE RIGHT VALUES
;
MOV      U. ACP(R4), R4   ; GET ACP STRUCTURE FROM PUD
ADD      #I. SY, R4       ; BUMP TO DEFAULT DEVICE STORAGE
CALL    .. GIAS          ;
BNE     40$                ; BR IF WE FOUND DEFAULT DEVICE
10$:    CMP      (SP), # "TO" ; DO WE WANT TO DEVICE ?
        BEQ     12$                ; BR IF YES
        CMP      (SP), # "TI"     ; OR TI DEVICE ?
        BNE     15$                ; BR IF NOT
12$:    MOV      #1, R5         ; RETURN A 1 FOR TI
        BR      40$                ; AND EXIT
15$:    CMP      (SP), # "OV"     ; WANT TO DO A TASK OVERLAY ?
        BNE     20$                ; BR IF NOT
        MOV      A. TD(R0), R5    ; GET STD FOR TASK
        MOV      S. DI(R5), R5    ; NOW GET DISK INDICATOR FOR TASK
        BIC      #177400, R5      ; IT'S A BYTE VALUE
        MUL      #U. SZ, R5       ; MULTIPLY BY SIZE OF PUD
        ADD      . PUDBA, R5     ; ADD BASE OF PUD
        BR      40$                ; AND RETURN PUD ADDRESS
20$:    MOV      . PUDBA, R5     ; MUST LOOK THROUGH PUD FOR A MATCH
25$:    CMP      U. DN(R5), (SP)  ; DID WE MATCH ?
        BNE     30$                ; BR IF NOT
        CMP      U. UN(R5), 2(SP) ; DO UNITS MATCH ?
        BEQ     40$                ; BR IF YES
30$:    ADD      #U. SZ, R5       ; BUMP TO NEXT PUD
        CMP      R5, . PUDEA     ; THROUGH PUD ?
        BLO     25$                ; TRY AGAIN IF NOT
        CMP      (SP)+, (SP)+    ; RESTORE STACK
        SEC     ; SHOW FAILURE
        BR      50$                ; AND SCRAM
40$:    CMP      (SP)+, (SP)+    ; RESTORE THE STACK
        CLR     R4                ; SHOW SUCCESS
50$:    RETURN                    ; RETURN TO CALLER
;
; . END      START

```

```

. TITLE SETDEF
. IDENT /MRH001/
. SBTTL INTRODUCTION
; PROGRAM SUPPORTING MRMC VARIABLE DEFAULT DEVICE PACKAGE
;
; PURPOSE OF PROGRAM IS TO CHANGE THE 'DEFAULT DEVICE' OF THE
; TERMINAL RUNNING THE PROGRAM
;
; CALLING SEQUENCE (FOR EXAMPLE)
;
; MCR>SDV SM:
;
; WOULD CHANGE THE 'DEFAULT DEVICE PUD POINTER', (LOCATED IN LOCATION
; U.LBH OF THE PUD FOR THE TT IN QUESTION) TO POINT TO THE PUD FOR THE
; DEVICE NAMED IN THE COMMAND LINE
;
; DUE TO THE SIMPLICITY OF THE COMMAND, NO INDIRECT LEVELS ARE SUPPORTED
; AUTHOR F. BORGER
; DEPT. OF MEDICAL PHYSICS
; MICHAEL REESE MEDICAL CENTER
; MARCH, 1976
;
. PAGE
. SBTTL DECODE DEVICE NAME
. MCALL GMCR$, QIOW$, EXIT$, DIR$
START: DIR$ #GETMCR ; GETMCR COMMAND LINE
MOV #GETMCR+G. MCRB, R0 ; START OF BUFFER -> R0
ADD #4, R0 ; BUMP PAST 'SDV '
CMPB (R0), #'A ; CHECK FOR ASCII CHARACTER
BLT BADSYN ; WOOPS, SYNTAX ERROR
CMPB (R0), #'Z ; I DON'T THINK I NEED TO COMMENT THIS !!
BGT BADSYN
CMPB 1(R0), #'A
BLT BADSYN
CMPB 1(R0), #'Z
BGT BADSYN
CMPB 2(R0), #15 ; TERMINATOR (IMPLYING DEVICE 0) ?
BEQ 2$ ; BR IF YES
CMPB 2(R0), #33 ; ALT MODE TERMINATOR (ALSO DEV ZERO) ?
BEQ 2$ ; BR IF YES
CMPB 2(R0), #' : ; DEVICE NUMBER A ':' ?
BNE 1$ ; BR IF NOT
2$: MOV #60, 2(R0) ; IF SO CHANGE IT TO AN ASCII '0'
1$: CMPB 2(R0), #'0 ; CHECK (ASCII) DEVICE NUMBER
BLT BADSYN
CMPB 2(R0), #'7
BGT BADSYN
FIND: BIC #60, 2(R0) ; ASCII NUMBER OK, CHANGE IT TO OCTAL
MOV . CRTSK, R1 ; MY ATL ADDRESS -> R1
MOV A. TI(R1), R1 ; GET MY TI PUD ADDRESS
MOV . PUDBA, R2 ; SET UP TO SCAN PUD FOR NEW DEFAULT DEVICE
CKPUD: CMP (R2), (R0) ; DO ASCII NAMES MATCH ?
BNE NOMAT ; BR IF NOT
CMPB 2(R2), 2(R0) ; NAMES MATCH, HOW 'BOUT UNIT NUMBERS ?
BNE NOMAT ; NO NO MATCH
; R2 NOW POINTS TO PUD OF DEVICE GUY AT TTY WANTS
MOV R2, R5 ; PUD POINTER -> R5
10$: MOV R5, U.LBH(R1) ; CHANGE DEFAULT DEVICE PUD POINTER
BR SCRAM ; AND SCRAM

```

```

;
;
NOMAT:  ADD      #U. SZ, R2                ; BUMP PUD POINTER TO NEXT PUD
        CMP      R2, . PUDEA              ; ARE WE STILL IN THE PUD ?
        BHIS     BAD                      ; NO, COULDN'T FIND DEVICE
        BR       CKPUD                   ; YES, TRY FOR MATCH AGAIN
        . PAGE
        . SBTTL  DECODE VOLUMN NAME
BADSYN: MOV      #NAMTAB, R4              ; GET NAME TABLE POINTER INTO R4
REGO:   CLR      R1                      ; FOR CAT5
        CALL     $CAT5                   ; CONVERT 1ST THREE CHARACTERS
        CMP      R1, (R4)                 ; NAMES MATCH ?
        BNE     NEXT                     ; BR IF NOT
        CMP      R2, #15                  ; TERMINATOR ?
        BEQ     OK                        ;
        CMP      R2, #33                  ; OR ESC ?
        BEQ     OK                        ;
        CLR     R1                        ; FOR CAT5
        CALL     $CAT5                   ; CONVERT NEXT THREE CHARACTERS
        CMP      R1, 2(R4)                ; NAMES MATCH ?
        BNE     NEXT                     ; BR IF NOT
        CMP      R2, #15                  ; TERMINATOR ?
        BEQ     OK                        ;
        CMP      R2, #33                  ; OR ESC ?
        BEQ     OK                        ;
        CLR     R1                        ; FOR CAT5
        CALL     $CAT5                   ; CONVERT 3RD THREE CHARACTERS
        CMP      R1, 4(R4)                ; NAMES MATCH ?
        BNE     NEXT                     ; BR IF NOT
        CMP      R2, #15                  ; TERMINATOR ?
        BEQ     OK                        ;
        CMP      R2, #33                  ; OR ESC ?
        BEQ     OK                        ;
        CLR     R1                        ; FOR CAT5
        CALL     $CAT5                   ; CONVERT LAST THREE CHARACTERS
        CMP      R1, 6(R4)                ; NAMES MATCH ?
        BEQ     OK                        ; BR IF OK
NEXT:   ADD      #14, R4                  ; BUMP TO NEXT ENTRY
        MOV      #GETMCR+G. MCRB+4, R0    ; RESET R0 TO START OF NAME
        TST     (R4)                      ; END OF TABLE ?
        BEQ     BAD                      ; BR IF SO
        BR      REGO                     ; ELSE SEARCH AGAIN
OK:     MOV      R4, R0                   ; POINT TO NEW ASCII DEVICE STRING
        ADD     #10, R0
        JMP     FIND                      ; AND FIND DEVICE PUD
BAD:    DIR$    #SYNBAD                   ; REPORT SYNTAX ERROR
SCRAM:  DIR$    #EXITPG                   ; EXIT, CPU LEFT
;
;
        . PAGE
        . SBTTL  DIRECTIVES AND DATA
;
; DIRECTIVE TO GET MCR COMMAND LINE
GETMCR: GMCR$                                ; GET MCR, (INCLUDES BUFFER)
;
;
; DIRECTIVE TO REPORT BAD SYNTAX
;
SYNBAD: QIOW$    IO. WVB, 5, 1, , IOST, , <SYNMES, SYNLEN, 40>

```

```

;
; ERROR MESSAGE
SYNMES: . ASCII <15>/***SDV ---SYNTAX ERROR/
        SYNLEN=. -SYNMES
        . EVEN
;
IOST:   . WORD    0,0
;
; EXIT DIRECTIVE
EXITPG: EXIT$
;
; TABLE OF VOLUME NAMES WITH EQUIVALENT DEVICES
;
NAMTAB: . RAD50   /MRH/
        . RAD50   /V62/
        . WORD    0,0
        . ASCII   /DP1:/
        . RAD50   /IAS/
        . RAD50   /V31/
        . WORD    0,0
        . ASCII   /DR0:/
        . RAD50   /SYS/
        . RAD50   /TWO/
        . WORD    0,0
        . ASCII   /DR2:/
        . RAD50   /IAS/
        . RAD50   /UPD/
        . WORD    0,0
        . ASCII   /DR3:/
        . WORD    0                ; END OF TABLE INDICATOR
        . END      START

```

---

```

DEVPAT/PR/-CP/-FP, DEVPAT/-SP=DEVPAT
LB: [1, 1]EXEC. STB/SS
/
PRI=240
UNITS=0
ACTFIL=0
/

```

---

```

SDV/PR/-FP, SDV/-SP=SDV, LBO: [1, 1]EXEC. STB
/
TASK= . . . SDV
PRI=222
ASG=TI: 5
STACK=32
//

```



## TOPICS

Section 1.0 Describes additional hardware supported by IAS V3.2.

Section 2.0 Describes new Software

Section 3.0 Describes new changes in IAS V3.2's EXEC

Section 4.0 Describes new and changed functionality for the Terminal Handler IAS V3.2. Examples of the new QIO's are provided.

Section 5.0 Node Pool Extension for Send Receive Directives

Section 6.0 Describes new changes in IAS V3.2's Program Development System (PDS)

Section 7.0 Describes new changes in IAS V3.2's Task Builder TKB

Section 8.0 IAS V3.2 Shadow Recording

## Section 1.0 NEW HARDWARE

This section describes the new central processors and peripheral devices supported by IAS V3.2.

### 1.1 NEW CENTRAL PROCESSORS

- PDP-11/23            The PDP-11/23 central processor uses the 18 bit LSI-11 Bus, has memory management and has a maximum of 124 KW.
- PDP-11/23 PLUS     The PDP-11/23 PLUS central processor uses the LSI-11 Bus, has a maximum of 4MB physical memory. The RLV22 controller is the only supported disks for the extended addressing.
- PDP-11/24            The PDP-11/24 system supports a variety of configurations. The PDP-11/24 CPU has 22-bit UNIBUS addressing capability and IAS supports a maximum of 4MB of physical memory.

### 1.2 PERIPHERAL DEVICES

IAS V3.2 supports the following new peripheral devices:

Device	Mnemonic	Type
RA60	DU	The RA60 is a high capacity, low cost, removable media disk.
RA80	DU	The RA80 is a high speed, fixed media disk with the UDA50 intelligent disk controller.
RA81	DU	The RA81 is a fixed media, Winchester-type disk.
TS11	MS	The TS11 is a 9-track magnetic tape storage system that connects to the UNIBUS. The recording density is 1600 BPI (PE) at 45 in/s. The TS11 was designed to operate at medium speeds in moderate-usage environments.

## Section 3.0 EXEC

This section covers the new directives added to the EXEC.

### 3.1 CNCT\$ TNAME, [EFN], [EAST], [ESB]

The connect directive synchronizes the task issuing the directive, with the exit or emit status of another active task (offspring). Execution of this directive creates linkage between the two task. The task that issues the directive becomes the Parent of the task connected to.

Multiple connects to a single task are not currently supported. Since Spawn automatically connects, you cannot connect to a spawned task.

- TNAME - The radix-50 name for the connected offspring.
- EFN - The event flag to be cleared when the directive is issued and set when the offspring task exits or emits status.
- EAST - The address of the Asynchronise Trap (AST) routine called when the offspring task exits or emits status.
- ESB - The one-word status block address written in when the offspring task exits or emits status.

### 3.2 EMST\$ [TNAME], STATUS

The Emit Status directive returns the specified 16-bit quantity to the specified connected task. It possibly sets an event flag or declares an AST if previously specified by the connected task in a Spawn or a Connect directive. In any case, whenever status is emitted to one or more tasks, those tasks no longer remain connected to the task issuing the Emit Status directive.

TNAME - is the radix-50 name of the task awaiting status.

STATUS is a 16-bit value to be returned to the connected task.

## Section 2.0 NEW SOFTWARE

### 2.1 NEW AND CHANGED SOFTWARE

A description of the new and changed software for IAS V3.2.

#### 2.1.1 Bad Block Utility (BAD)

We have included a new version of the BAD block utility (BAD). The new BAD requires that the input disc be mounted foreign.

#### 2.1.2 Privileged Backup and Restore Utility (BRUPR)

A privileged Backup and Restore utility (BRUPR) is included. Functionally identical to BRU, it is built with the /PR:0 switch. This allows the input disk to be mounted during the copy operation.

#### 2.1.3 Editor (EDT)

EDT V3. has been added to IAS V3.2.

#### 2.1.4 Disc volume formatter (FMT)

FMT now requires the input disk to be mounted foreign before it can be formatted.

#### 2.1.5 INItialize

The disk volume initialization task requires that any volume to be initialized is first mounted foreign.

#### 2.1.6 MACro

A new version of the PDP-11 MACRO-11 assembler is included with IAS V3.2.

## Section 4. New Functionality for the Terminal Handler (QIO's)

### 4.1 Auto Baud detection Introduction

The Terminal handler will now automatically detect the baud rate of any incoming call (dialup lines only). The handler samples the input character, determines the incoming baud rate and sets the interface speed accordingly.

#### 4.1.1 WHAT TO EXPECT

When a user dials in, one "WILL NOT" receive the customary PDS WELCOME (time Sharing) message or the customary MCR display. The remote terminal will not display anything at first, a user must TYPE a Carriage Return multiple times so that the incoming callers baud rate can be determined. Then the system will send the customary banner message. On a Timesharing System if the user types more carriage returns than are necessary for autobaud detection the Username prompt will be redisplayed.

#### 4.1.2 HOW TO AUTO-BAUD

The TER command allows a user logged under [1,1] to change the characteristics of a specified terminal. Auto-baud can be achieved via one of the following commands.

MCR interface -

```
MCR>TER Ttn:/AUTO
```

SCI or PDS interface -

```
PDS> SET TERM:Ttn: AUTOBAUD
```

### 4.2 NEW TERMINAL CHARACTERISTICS

New terminal characteristics are included in V3.2's TT (Terminal handler). These characteristics are:

TC.8BC	Pass eight bits for all read QIO'S
TC.NEC	No echo
TC.ANI	Terminal supports ANSI escape sequences
TC.AVO	Terminal has an advanced video option
TC.DEC	Terminal is DEC VT100 compatible
TC.EDT	Terminal supports VT102 type editing escape sequences
TC.RGS	Terminal supports REGIS (VT125 type) commands
TC.ABD	Terminal line automatically determines caller's baudrate

Keywords:

Keywords have been added to V3.2 to allow modification of the above characteristics using the MCR TER Ttn:/keyword or the SCI/PDS SET TER:Ttn: keyword commands. The following table lists the new keywords.

Keywords	Characteristic
EIGHT_BIT_CHARACTERS	TC.8BC
ECHO	TC.NEC
ANSI_CRT	TC.ANI
ADVANCED_VIDEO	TC.AVO
DEC_CRT	TC.DEC
EDIT	TC.EDT
REGIS	TC.RGS
AUTOBAUD	TC.ABD

The TT handler also supports the following new terminal types.

LA120  
LA34  
LA38  
LA100  
LA180  
VT61  
VT101  
VT102  
VT105  
VT125  
VT131  
VT132  
VT2XX (VT220,VT240,VT241)

Three I/O subfunctions (QIO) have been added to the terminal handler:

4.2. QIO IO.ATA,...<[AST],[PARAMETER2][,[AST2]]>

Example:

```
QIO$S #IO.ATA,#TILUN,#1,...,<#DFAST,,#CNTRLC> ;attach to LUN
BCS ERR1 ; Error 1 IO.ATA error for DFAST
```

IO.ATA - is an ATTACH function and specifies asynchronous system traps (ASTs) to process unsolicited TT input. Control passes to the AST routine when the task receives an unsolicited character (other than CTRL/Q, CTRL/S, CTRL/X, or CTRL/O). When specifying a CTRL/C AST on a timesharing system, the AST will be checked for the required privileges before the AST address is passed to the TCP (Timesharing Control Primitives). Thus, it is not necessary to include the CTC\$T call to TCP in your application task. When you detach your terminal, TCP will cancel the CTRL/C AST.

AST - specifies the entry point for a routine that is entered when an unsolicited character other than CTRL/Q, CTRL/S, CTRL/X, or CTRL/O is received.

PARAMETER2 - identifies a terminal in a multi-terminal environment. It is placed into the high byte of the first word on the stack (SP+00) when the trap occurs. The low byte contains the unsolicited character.

AST2 - specifies the entry point for a routine that is entered when an unsolicited CTRL/C is received.

If one or more parameters are omitted, the function reverts to that supported in V3.1.

The subfunctions TF.NOT and TF.XCC are also new.

Symbolic name	Subfunction
TF.NOT	unsolicited input AST notification; Unsolicited characters are stored in the type-ahead buffer until they are read by the task.
TF.XCC	Exclude CNTL/C from the unsolicited characters passed to the task. CNTL/C will instead be passed to the user's CLI (MCR or PDS)

Setting the conditional assembly parameter U\$\$CHA in [311,114]PARAMS.MAC allows support for this QIO.

#### 4.2.1 QIO IO.RST,...,<STADD, SIZE [,TMO]>

IO.RST - reads characters from a TT until the input buffer is filled or a special character (in the ranges 0-037 or 171-177) is received.

STADD - is the address of the receiving buffer.

SIZE - is the buffer length in bytes.

TMO - is an optional time-out count in 10-second intervals for the full-duplex driver. If 0 is specified, no time-out can occur. Time-out is the maximum time allowed between two input characters before the read is aborted.

Setting the conditional assembly parameter I\$\$RST in [311,114]PARAMS.MAC allows support for this qio.

#### 4.3. QIO IO.RTT,....,<STADD, SIZE [,TMO], TABLE>

IO.RTT reads characters from a TT until the input buffer is filled or a user-specified character (in the range 0-377) is received.

STADD is the address of the receiving buffer.

SIZE is the buffer length in bytes.

TMO is an optional time-out count in 10-second intervals for the full-duplex driver. If 0 is specified, no time-out can occur. Time-out is the maximum time allowed between two input characters before the read is aborted.

TABLE is the address of a sixteen-word table that specifies the end-of-read characters. Each bit in the table represents an ASCII character. The first word represents the ASCII character codes 0-17. The bits of the second word represent the ASCII codes 20-37, and so forth.

This support is a conditional assembly specified by setting the parameter I\$\$RST in [311,114]PARAMS.MAC.



## Section 5. Node Pool Extension for Send Receive Directives

You can generate a system with an up to 4 KW extension to the Send/Receive node pool. The method is to use a 4 KW user controlled partition named SENPAR.

You can generate a system that can extend SEND/RECEIVE node Pool by using a user controlled Partition named SENPAR. SENPAR may be as large as 4 Kw and is used as an alternative to node pool. The partition name MUST be SENPAR and it must be first in the Task Partition Directory. This makes it the first partition to be declared to Sysgen Phase one (SGN1). Unused parts of the TPD entry are reused to hold extra parameters. The executive will use nodes from SENPAR for all SEND and RECEIVE requests.

## Section 6. Program Development System (PDS)

Several changes have been made to PDS for IAS V3.2. The details of these changes are discussed below.

### 6.1 LOGIN

The Login precedure has been simplified. PDS now prompts for the user's name when it is activated on a logged out terminal.

### 6.2 SET Changes

SET [NO]SCI - allows you to execute SCI commands from a non-console terminal. This SET requires SCI privileges. The SHUTDOWN and BOOT commands are not allowed.

SET [NO]ECHO - turns ON/OFF the echo of commands from indirect command files. PDS does an internal SET ECHO when the command file is finished.

SET [NO]VERIFY same as SET [NO]ECHO

SET [NO]LOGIN Enables/disables LOGIN commands.

### 6.3 LOGIN.CMD

User login command files execute in NO ECHO mode.

### 6.4 FOR/77

The F77 qualifer added to IAS is the same as the F4P qualifer, but calling F77.

### 6.5 PURGE

The PURGE command is equivalent to PDS>DELETE/KEEP.

### 6.6 MOU/OVERRIDE:IDENTIFICATION

PDS has been modified to accept the /OVERRIDE:IDENTIFICATION (override volume identification) qualifier as a synonym for the /OVERRIDE:VOLUME\_ID qualifier. Also a dummy volume name is no longer required.

#### NOTE

LOGIN/LOGOUT messages are written by PDS to the device assigned to LUN 13. The LUN may be reassigned if desired.

## Section 7.0 Task Builder TKB

Several features have been added to TKB for IAS V3.2.

### 7.1 Switches

A new option has been added to the Task Image Switch /PR (allow all privileged access and I/O operations), as follows:

/PR allows all privileged operations and maps the EXEC.

/PR:0 allows all privileged operations and does not map the EXEC.

### 7.2 Autoload Vector PSECT

The Autoload Vector PSECT is now generated with the read-only characteristic. This may prevent some tasks from building. TKB will issue the error message "NOT ENOUGH APRS TO MAP TASK". The task may be built by using the /RW (read-write) qualifier on the task image name.

### 7.3 File Structure Verification Utility (VFY)

A new version of VFY is included in this distribution kit.

#### FILE STRUCTURE VERIFICATION UTILITY (VFY)

The File Structure Verification Utility (VFY) for Files-11 volumes provides the ability to:

- o Check the readability and validity of a file-structured volume (default function).
- o Print the number of available blocks on a file-structured volume (/FR).
- o Search for files in the index file that are not in any directory; that is, files that are "lost" in the sense that they cannot be accessed by file name (/LO). (See the IAS/RSX-11 I/O Operations Reference Manual for a description of the index file).
- o Validate directories against the files they list (/DV).
- o List all files in the index file, showing the file ID, file name, and owner (/LI).
- o Mark as "used" all the blocks that appear to be available but are actually allocated to a file (/UP).
- o Rebuild the storage allocation bitmap so that it properly reflects the information in the index file (/RE).

- o Restore files that a marked for deletion (/DE).
- o Delete bad file headers (/HD).
- o Perform a read check on every allocated block on a file-structured volume (/RC).

The volume to be verified must be mounted as a Files-11 device.

## Section 8.0 IAS V3.2 Shadow Recording

Shadow Recording has been added as an optional feature to IAS V3.2 to enhance data integrity.

Shadow recording is a technique used to increase data availability by redundantly recording information nearly simultaneously on two disks of the same type. The redundancy and recovery of the information is transparent to user-level tasks.

The two disks used when Shadow Recording is in effect are called the Shadow set. The disk being shadowed is called the primary disk, the shadow disk is called the secondary disk.

During Shadow Recording, the IAS Executive duplicates each Write I/O Packet to a primary disk and directs it to the secondary disk of the shadow set.

During a read from a shadow set, the IAS Executive reads the primary disk first. If a read error occurs on the primary disk, the Executive reads the secondary disk.

If the read is then successful, the information is returned to the operator along with status showing that the information is good.

See the IAS System Generation and Management Guide for detailed information on the use of the shadow recording feature.

The advantages of using Shadow Recording are:

- o Near simultaneous data recording and backup.
- o Quicker restoration of data.
- o Decreased likelihood of a hard disk read error.

The DISADVANTAGES of using Shadow Recording are:

- o Doubles the number of I/O Write Packets.
- o Takes up space in the Executive pool space.
- o Decreases throughput.
- o Uses an entire disk drive.

The Shadow Recording feature is added to the system during system generation.

## 8.1 Shadow Recording Control

Shadow Recording is controlled by five sub-commands, listed and described in the following section. You gain access to these these sub-commands with the SHADOW command in one of the following two formats:

>SHAdow sub-command

Or, you may enter the command on two lines by invoking the Shadow Recording task first. For example:

>SHA

### 8.1.1 Shadow Recording Sub-Commands

You can use five commands to control Shadow Recording. However, you can only enter these commands from a privileged terminal. The commands allow you to start, stop, abort, or continue Shadow Recording, and to display all device pairs that are being shadowed.

#### 8.1.2 ABORT Command

SHA>ABORT ddnn:

ddnn: The device-unit of the primary Files-11 device.

The ABORT command stops Shadow Recording even if catch-up is in progress. The ABORT command:

- o Verifies that the primary disk is part of a shadowed pair
- o Aborts catch-up if it is in progress and then executes a STOP command

#### 8.1.3 CONTINUE Command

SHA>CONTINUE ddnn: TO ddxx:

ddnn The primary Files-11 device that is to be duplicated on the secondary Files-11 device.

IAS Shadow Recording Notes

---

ddxx      The secondary device that is to be the shadowed copy of the primary device.

The CONTINUE command allows you to restart Shadow Recording on the same two disks that you were using when the STOP command was issued. The CONTINUE command assumes that the two disks are physically alike and it does not check the disks for equality. This means that after you issue the STOP command you must be sure that no further writing occurs on the primary or secondary disks.

The CONTINUE command:

- o Assumes that the primary and secondary disks are identical
- o Verifies that the primary disk is a Files-11 pack and mounted properly
- o Verifies that the secondary is mounted foreign
- o Verifies that both disks are identical disk types
- o Sets up the Shadow Recording data structure, which starts the Shadow Recording operation within the Executive
- o Starts the copying of the primary disk to the secondary disk (catch-up)

SHA>DISPLAY

The DISPLAY command shows all shadowed pairs. Entering the DISPLAY command causes a display of the shadowed device pairs in the following format:

SCB	PRIMARY	SECONDARY
xxxxxx	ddnn:	ddxx:
xxxxxx	The address of the SCB control block for Shadow Recording. This control block is in the system Executive pool space.	
ddnn	The primary Files-11 device that is to be duplicated on the secondary Files-11 device.	
ddxx	The secondary device that is to be the shadowed copy of the primary device.	

## IAS Shadow Recording Notes

---

### 8.1.4 START Command

SHA>START ddnn: TO ddx:

ddnn      The primary Files-11 device that is to be duplicated on the secondary Files-11 device.

ddxx      The secondary device that is to be the shadowed copy of the primary device. The START command:

- o Verifies that the primary disk is a Files-11 device and mounted
- o Verifies that the secondary disk is mounted as a foreign device.
- o Verifies that both the primary and secondary disks are identical disk types
- o Set up the Shadow Recording data structure (SCB), which starts the Shadow Recording operations within the Executive.
- o Starts the copying of the primary disk to the secondary disk (catch-up).

After the START command initiates Shadow Recording, all data written to the primary disk is also written to the secondary disk. Since Shadow Recording is implemented by duplicating I/O Write Packets, the secondary disk may not logically duplicate the primary disk at any given time.

The information about the last block copied is in the SCB control block.

A fully redundant secondary disk does not exist until the catch-up operation has completed.

### 8.1.5 STOP Command

STOP ddnn:

ddxx:      The device-unit of the primary device.



## IAS Shadow Recording Notes

---

The STOP command prevents Shadow Recording from continuing, unless catch-up is in progress. If you issue the STOP command during catch-up, you will receive an error message.

The absence of outstanding I/O does not necessarily mean that all file activity has stopped. You must make certain that all tasks that write onto the shadowed pair have stopped before you issue STOP or ABORT to ensure the integrity of the secondary data.

The STOP command:

- o Verifies that the primary device is part of a shadowed pair
- o Unlinks the SCB data structure, if no outstanding I/O exists, thereby stopping the Executive from Shadow Recording.
- o Marks the data structure for deletion if outstanding I/O does exist

### 8.1.6 Shadow Recording Restrictions

- o A primary and a secondary disk may not overlap
- o Both disks must be of the same device type

## KNOWN RESTRICTIONS IN IAS V3.2

Do not use the BAD utility on the bootable tape. The one on the system disc should be used instead.

## RESTRICTIONS LIFTED FOR V3.2

The main purpose of IAS V3.2 is to correct errors, both in the software and in the manual set. IAS V3.2 contains corrections to all problems for which patches were published in the IAS V3.1 issues of the Software Dispatch to date. Additionally all corrections to IAS V3.1 supplied on the IAS Autopatch Update have been included in IAS V3.2.