



```

EEEEEEEEEE XX      XX  CCCCCCCC DDDDDDDD BBBB8888 GGGGGGGG
EEEEEEEEEE XX      XX  CCCCCCCC DDDDDDDD BBBB8888 GGGGGGGG
EE          XX      XX  CC          DD      DD  BB      BB  GG
EE          XX      XX  CC          DD      DD  BB      BB  GG
EE          XX      XX  CC          DD      DD  BB      BB  GG
EEEEEEEEEE XX      XX  CCCCCCCC DDDDDDDD BBBB8888 GG
EEEEEEEEEE XX      XX  CCCCCCCC DDDDDDDD BBBB8888 GG
EE          XX      XX  CC          DD      DD  BB      BB  GG GGGGGG
EE          XX      XX  CC          DD      DD  BB      BB  GG GGGGGG
EE          XX      XX  CC          DD      DD  BB      BB  GG GG
EEEEEEEEEE XX      XX  CCCCCCCC DDDDDDDD BBBB8888 GGGGGG
EEEEEEEEEE XX      XX  CCCCCCCC DDDDDDDD BBBB8888 GGGGGG

```

```

LL          IIIIII SSSSSSSS
LL          IIIIII SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLL IIIIII SSSSSSSS

```



```
48 0047 1 ! Include files:
49 0048 1 !
50 0049 1 MACRO $module_name string = 'exch$dbg' %; . The require file needs to know our module name
51 0050 1 REQUIRE 'SRC$:EXCREQ'
52 0051 1 ;
53 0148 1
54 0149 1 PSECT
55 0150 1     OWN = exch$dbg_own (WRITE)
56 0151 1 ;
57 0152 1
58 0153 1 %SBTTL 'Module table of contents'
59 0154 1
60 0155 1 ! Module table of contents:
61 0156 1 !
62 0157 1 FORWARD ROUTINE
63 0158 1     exch$dbg_dump_dosv : NOVALUE, ! Format the contents of a DOS11$ block
64 0159 1     exch$dbg_fab_dev_print, ! Print device characteristics from FAB info
65 0160 1     exch$dbg_fab_dump, ! Hex dump of a fab
66 0161 1     exch$dbg_fab_name_print, ! Print the name information attached to a FAB
67 0162 1     exch$dbg_nam_dump, ! Hex dump of a nam
68 0163 1     exch$dbg_utl_print_devchar ! Explode a DEV device characteristics field (e.g. FAB$$_DEV
69 0164 1 ;
70 0165 1
71 0166 1 ! EXCHANGE facility routines
72 0167 1 !
73 0168 1 EXTERNAL ROUTINE
74 0169 1     exch$dos11_expand_filename, ! Expand filename info into a context block
75 0170 1     exch$util_dos11ctx_allocate, ! Grab a context block
76 0171 1     exch$util_dos11ctx_release : NOVALUE ! Return the context block
77 0172 1 ;
78 0173 1
79 0174 1 ! Bound declarations:
80 0175 1 !
81 0176 1 ! BIND
82 0177 1 ! ;
```

```

: 84 0178 1 GLOBAL ROUTINE exch$dbg_dump_dosv (volb : $ref_bblock, flags) : NOVALUE =
: 85 0179 2 BEGIN
: 86 0180 2 $dbgtrc_prefix ('dbg_dump_dosv> ');
: 87 0181 2
: 88 0182 2 LOCAL
: 89 0183 2     ctx : $ref_bblock,
: 90 0184 2     ent : $ref_bblock,
: 91 0185 2     lcl : $bblock [4],
: 92 0186 2     first
: 93 0187 2 ;
: 94 0188 2
: 95 0189 2 BIND
: 96 0190 2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock
: 97 0191 2 ;
: 98 0192 2
: 99 0193 2 BUILTIN
100 0194 2     ACTUALCOUNT
101 0195 2 ;
102 0196 2
103 0197 2 IF ACTUALCOUNT () EQL 2
104 0198 2 THEN
105 0199 2     lcl [0,0,32,0] = .flags
106 0200 2 ELSE
107 0201 2     lcl [0,0,32,0] = -1;
108 0202 2
109 0203 2 IF .lcl [dmpdsv$v_volb]
110 0204 2 THEN
111 0205 2     $print_fao ('    dump_dosv> entry - volb !XL, dosv !XL', .volb, .dosv);
112 0206 2
113 0207 2 $block_check (2, .volb, volb, 623);
114 0208 2 $block_check (2, .dosv, dos11, 624);
115 0209 2
116 0210 2 IF .lcl [dmpdsv$v_status]
117 0211 2 THEN
118 P 0212 2     $print_fao ('    dump_dosv> status - posvalid !UL, bot !UL, tapemark !UL, eot !UL, dir !UL, curmrkd !UL'
119 P 0213 2         .dosv [dos11$v_position_valid], .dosv [dos11$v_beg_of_tape], .dosv [dos11$v_tape_mark],
120 0214 2         .dosv [dos11$v_end_of_tape], .dosv [dos11$v_directory], .dosv [dos11$v_current_marked]);
121 0215 2
122 0216 2 IF .lcl [dmpdsv$v_position]
123 0217 2 THEN
124 P 0218 2     $print_fao ('    dump_dosv> .dos11$a_current_entry !XL, .dos11$l_current_file !UL',
125 0219 2         .dosv [dos11$a_current_entry], .dosv [dos11$l_current_file]);
126 0220 2
127 0221 2 IF .lcl [dmpdsv$v_entries]
128 0222 2 THEN
129 0223 2     BEGIN
130 0224 2
131 0225 2     ! If the pointer points to the queue header, we have reached the end of the list
132 0226 2     !
133 0227 2     ent = .dosv [dos11$a_entry_flink];
134 0228 2     ctx = exch$util_dos11ctx_allocate (.volb, 0);           ! Grab a context block for the loop
135 0229 2     first = true;
136 0230 2     WHILE .ent NEQ dosv [dos11$q_entry_header]
137 0231 2     DO
138 0232 2         BEGIN
139 0233 2             IF .first
140 0234 2
```





7E	60	50	0C	A7	9E	0006A	MOVAB	12(R7), R0	0214		
7E	60	01		05	EF	0006E	EXTZV	#5, #1, (R0), -(SP)			
7E	60	01		04	EF	00073	EXTZV	#4, #1, (R0), -(SP)			
7E	60	01		03	EF	00078	EXTZV	#3, #1, (R0), -(SP)	67		
7E	60	01		02	EF	0007D	EXTZV	#2, #1, (R0), -(SP)			
7E	60	01		01	EF	00082	EXTZV	#1, #1, (R0), -(SP)			
7E	60	01		00	EF	00087	EXTZV	#0, #1, (R0), -(SP)			
			0000'	CF	9F	0008C	PUSHAB	P.AAC			
	00000000G	EF		07	FB	00090	CALLS	#7, EXCH\$UTIL_FA0_BUFFER			
				50	DD	00097	PUSHL	R0			
		6B		01	FB	00099	CALLS	#1, LIB\$PUT_OUTPUT			
	16	53		02	E1	0009C	BBC	#2, LCL, 5\$	0216		
			0E	A7	DD	000A0	PUSHL	14(R7)	0219		
			1A	A7	DD	000A3	PUSHL	26(R7)			
			0000'	CF	9F	000A6	PUSHAB	P.AAE			
	00000000G	EF		03	FB	000AA	CALLS	#3, EXCH\$UTIL_FA0_BUFFER			
				50	DD	000B1	PUSHL	R0			
		6B		01	FB	000B3	CALLS	#1, LIB\$PUT_OUTPUT			
	01	53		03	E0	000B6	BBS	#3, LCL, 6\$	0221		
				04		000BA	RET				
		56	12	A7	D0	000BB	MOVL	18(R7), ENT	0227		
				7E	D4	000BF	CLRL	-(SP)	0228		
			04	AC	DD	000C1	PUSHL	VOLB			
	00000000G	EF		02	FB	000C4	CALLS	#2, EXCH\$UTIL_DOS11CTX_ALLOCATE			
		58		50	DD	000CB	MOVL	R0, CTX			
		59		01	DD	000CE	MOVL	#1, FIRST	0229		
		5A	66	A8	9E	000D1	MOVAB	102(R8), R10	0250		
		50	12	A7	9E	000D5	MOVAB	18(R7), R0	0230		
		50		56	D1	000D9	CPL	ENT, R0			
				74	13	000DC	BEQL	9\$			
		33		59	E9	000DE	BLBC	FIRST, 8\$	0234		
			0000'	CF	9F	000E1	PUSHAB	P.AAG	0237		
		6B		01	FB	000E5	CALLS	#1, LIB\$PUT_OUTPUT			
		7E	63	8F	9A	000E8	MOVZBL	#99, -(SP)	0240		
			000F423F	8F	DD	000EC	PUSHL	#999999			
			0000'	CF	9F	000F2	PUSHAB	P.AAK			
				0C	DD	000F6	PUSHL	#12			
		7E	12	A7	7D	000F8	MOVQ	18(R7), -(SP)			
			12	A7	9F	000FC	PUSHAB	18(R7)			
		7E	270F	8F	3C	000FF	MOVZWL	#9999, -(SP)			
			0000'	CF	9F	00104	PUSHAB	P.AAI			
	00000000G	EF		09	FB	00108	CALLS	#9, EXCH\$UTIL_FA0_BUFFER			
				50	DD	0010F	PUSHL	R0			
		6B		01	FB	00111	CALLS	#1, LIB\$PUT_OUTPUT			
				59	D4	00114	CLRL	FIRST	0242		
		3C	A8	08	A6	14	28	00116	MOV3	#20, 8(ENT), 60(CTX)	0244
				58	DD	0011C	PUSHL	CTX	0245		
			00000000G	EF	01	FB	0011E	CALLS	#1, EXCH\$DOS11_EXPAND_FILENAME		
7E	1A	A6		00	EF	00125	EXTZV	#0, #1, 26(ENT), -(SP)	0250		
				7E	16	A6	3C	0012B	MOVZWL	22(ENT), -(SP)	
						5A	DD	0012F	PUSHL	R10	
					50	A8	DD	00131	PUSHL	80(CTX)	
		7E			66	7D	00134	MOVQ	(ENT), -(SP)		
					56	DD	00137	PUSHL	ENT		
		7E	18	A6	3C	00139	MOVZWL	24(ENT), -(SP)			
			0000'	CF	9F	0013D	PUSHAB	P.AAL			
	00000000G	EF		09	FB	00141	CALLS	#9, EXCH\$UTIL_FA0_BUFFER			

67

20

41

20

0221

0227

0228

0229

0250

0230

0234

0237

0240

0242

0244

0245

0250



EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
Module table of contents

6 5  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 7  
(3)

EXC  
V04

		50	DD	00148	PUSHL	R0	:
	6B	01	FB	0014A	CALLS	#1, LIB\$PUT_OUTPUT	:
	56	66	DD	0014D	MOVL	(ENT), ENT	: 0252
		83	11	00150	BRB	7\$	: 0230
		58	DD	00152	9\$: PUSHL	CTX	: 0255
00000000G	EF	01	FB	00154	CALLS	#1, EXCH\$UTIL_DOS11CTX_RELEASE	:
		04	0015B	RET			: 0259

; Routine Size: 348 bytes, Routine Base: EXCH\$DBG\_CODE + 0000

: R

```
167 0260 1 GLOBAL ROUTINE exch$dbg_fab_dev_print (fabb : $ref_bblock) = %SBTTL 'exch$dbg_fab_dev_print'
168 0261 2 BEGIN
169 0262 2
170 0263 2 $dbgtrc_prefix ('fab_dev_print > ');
171 0264 2
172 0265 2 OWN
173 0266 2     devbufsiz,
174 0267 2     devchar,
175 0268 2     devclass,
176 0269 2     devdepend,
177 0270 2     devnam      : $bblock [64],
178 0271 2     devnamlen,
179 0272 2     devtype,
180 0273 2     maxblock;
181 0274 2
182 0275 2 OWN
183 0276 2     dev_item : VECTOR [22, LONG] INITIAL (
184 0277 2         WORD (dvi$ devbufsiz, 4),           LONG (devbufsiz,0),
185 0278 2         (dvi$ devchar^16 OR 4),           devchar,      0,
186 0279 2         (dvi$ devclass^16 OR 4),         devclass,     0,
187 0280 2         (dvi$ devdepend^16 OR 4),       devdepend,   0,
188 0281 2         (dvi$ fulldevnam^16 OR 64),     devnam,      devnamlen,
189 0282 2         (dvi$ devtype^16 OR 4),        devtype,     0,
190 0283 2         (dvi$ maxblock^16 OR 4),       maxblock,    0,
191 0284 2         0);
192 0285 2
193 0286 2 LOCAL
194 0287 2     status,
195 0288 2     inp_desc : $desc_block,
196 0289 2     tmp_desc : $desc_block,
197 0290 2     nam      : $ref_bblock;           ! Pointer to the name block
198 0291 2
199 0292 2     nam = .fabb [fab$l nam];         ! Get pointer to the name block
200 0293 2     $stat_str_desc_init (inp_desc, .nam [nam$b_dev], .nam [nam$l_dev]); ! Input Device name address
201 0294 2
202 0295 2     status = $getdviw (efn=0, devnam=inp_desc, itmlst=dev_item);
203 0296 2     IF NOT .status
204 0297 2     THEN
205 0298 2     BEGIN
206 0299 2         $print_fao ('***** Error in $GETDVI for device '!AS' *****', inp_desc);
207 0300 2         status = ((.status AND %X'FFFFFFF8') OR 1);
208 0301 2         SIGNAL ($warning_stat (status));
209 0302 2         RETURN 0;
210 0303 2     END;
211 0304 2
212 0305 2     tmp_desc [dsc$w_length] = .devnamlen;
213 0306 2     tmp_desc [dsc$a_pointer] = devnam;
214 0307 2     $print_fao ('Getdvi for name '!AS' resolved to device '!AS'', inp_desc, tmp_desc);
215 0308 2     $print_fao ('Bufsiz = !UL, Maxblocks = !UL, Class = !XB, Type = !XB', .devbufsiz, .maxblock, .devclass, .
216 0309 2     BEGIN
217 0310 2     BIND
218 0311 2         dep = devdepend : $bblock,
219 0312 2         dev = devchar   : $bblock;
220 P 0313 2     $print_fao ('Cylinders = !UL, Tracks = !UL, Sectors = !UL, DevChar = !XL'
221 0314 2         .dep[0,16,16,0], .dep[0,8,8,0], .dep[0,0,8,0], .dev [0,0,32,0]);
222 0315 2
223 0316 2     END;
```

```

: 224      0317 2 exch$dbg_utl_print_devchar (.devchar);
: 225      0318 2
: 226      0319 2 RETURN 1;
: 227      0320 1 END;

```

.PSECT EXCH\$DBG\_OWN,NOEXE,2

00000 DEVBUFSIZ:

.BLKB 4

00004 DEVCHAR:.BLKB 4

00008 DEVCLASS:

.BLKB 4

0000C DEVDEPEND:

.BLKB 4

00010 DEVNAM: .BLKB 64

00050 DEVNAMLEN:

.BLKB 4

00054 DEVTYPE: .BLKB 4

00058 MAXBLOCK:

.BLKB 4

0004 0008 0005C DEV\_ITEM:

.WORD 8, 4

00000000' 00060 .ADDRESS DEVBUFSIZ

00000000' 00064 .LONG 0

00020004' 00068 .LONG 131076

00000000' 0006C .ADDRESS DEVCHAR

00040004' 00070 .LONG 0, 262148

00000000' 00078 .ADDRESS DEVCLASS

000A0004' 0007C .LONG 0, 655364

00000000' 00084 .ADDRESS DEVDEPEND

00E80040' 00088 .LONG 0, 15204416

00000000' 00090 .ADDRESS DEVNAM, DEVNAMLEN

00060004' 00098 .LONG 393220

00000000' 0009C .ADDRESS DEVTYPE

001A0004' 000A0 .LONG 0, 1703940

00000000' 000A8 .ADDRESS MAXBLOCK

00000000' 000AC .LONG 0, 0

.PSECT EXCH\$DBG\_PLIT,NOWRT,2

6E	69	20	72	6F	72	72	45	20	20	2A	2A	2A	2A	2A	001D4	P.AAO:	.ASCII	\***** Error in \$GETDVI for device '!AS'\
65	64	20	72	6F	66	20	49	56	44	54	45	47	24	20	001E3			
					22	53	41	21	22	20	65	63	69	76	001F2			
							00	2A	2A	2A	2A	2A	20	20	001FC			

010E002F' 00204 P.AAN: .ASCII \ \*\*\*\*\*\<0>

00000000' 00208 .LONG 17694767

00000000' 0020C .ADDRESS P.AAO

65	6D	61	6E	20	72	6F	66	20	69	76	64	74	65	47	0020C	P.AAQ:	.ASCII	\Getdvi for name '!AS' resolved to device\
64	65	76	6C	6F	73	65	72	20	22	53	41	21	22	20	0021B			
					65	63	69	76	65	64	20	6F	74	20	0022A			
							00	00	22	53	41	21	22	20	00234			

010E002E' 0023C P.AAP: .ASCII \ '!AS'\<0><0>

00000000' 00240 .LONG 17694766

00000000' 00244 .ADDRESS P.AAQ

20	20	2C	4C	55	21	20	3D	20	7A	69	73	66	75	42	00244	P.AAS:	.ASCII	\Bufsiz = !UL, Maxblocks = !UL, Class =\
4C	55	21	20	3D	20	73	6B	63	6F	6C	62	78	61	4D	00253			
					3D	20	73	73	61	6C	43	20	20	2C	00262			

```

21 20 3D 20 65 70 79 54 20 20 2C 42 58 21 20 0026C
00 00 00 42 58 0027B
010E0039 00280 P.AAR: .LONG 17694777
00000000' 00284 .ADDRESS P.AAS
4C 55 21 20 3D 20 73 72 65 64 6E 69 6C 79 43 00288 P.AAU: .ASCII \Cylinders = !UL, Tracks = !UL, Sectors =\
2C 4C 55 21 20 3D 20 73 6B 63 61 72 54 20 2C 00297
3D 20 72 61 68 43 76 65 44 20 2C 4C 55 21 20 002A6
00 4C 58 21 20 002B0 .ASCII \ !UL, DevChar = !XL\<0>
010E003B 002C4 P.AAT: .LONG 17694779
00000000' 002C8 .ADDRESS P.AAU

```

DEP=  
DEV=

DEVDEPEND  
DEVCHAR

.EXTRN SYSSGETDVIW

.PSECT EXCH\$DBG\_CODE,NOWRT,2

.ENTRY EXCH\$DBG FAB DEV PRINT, Save R2,R3,R4,R5

MOVAB LIB\$PUT\_OUTPUT, R5

MOVAB EXCH\$UTIL\_FAO\_BUFFER, R4

MOVAB DEP, R3

SUBL2 #16, SP

MOVL FABB, R0

MOVL 40(R0), NAM

MOVW #270, DESC+2

MOVZBW 57(NAM), DESC

MOVL 68(NAM), DESC+4

CLRQ -(SP)

CLRQ -(SP)

PUSHAB DEV\_ITEM

PUSHAB INP\_DESC

CLRQ -(SP)

CALLS #8, SYSSGETDVIW

MOVL R0, STATUS

BLBS STATUS, 1\$

PUSHAB INP\_DESC

PUSHAB P.AAN

CALLS #2, EXCH\$UTIL\_FAO\_BUFFER

PUSHL R0

CALLS #1, LIB\$PUT\_OUTPUT

BICL3 #7, STATUS, R0

BISL3 #1, R0, STATUS

BICB2 #7, STATUS2

PUSHL STATUS2

CALLS #1, LIB\$SIGNAL

BRB 2\$

MOVW DEVNAMLEN, TMP\_DESC

MOVAB DEVNAM, TMP\_DESC+4

PUSHL SP

PUSHAB INP\_DESC

PUSHAB P.AAP

CALLS #3, EXCH\$UTIL\_FAO\_BUFFER

PUSHL R0

CALLS #1, LIB\$PUT\_OUTPUT

PUSHL DEVTYPE

PUSHL DEVCLASS

```

: 0260
:
: 0292
: 0293
: 0295
:
: 0296
: 0299
:
: 0300
: 0301
:
: 0302
: 0305
: 0306
: 0307
:
: 0308
:

```

	4C	A3	DD	0008E	PUSHL	MAXBLOCK	
	F4	A3	DD	00091	PUSHL	DEVBUFSIZ	
	0000'	CF	9F	00094	PUSHAB	P.AAR	
64		05	FB	00098	CALLS	#5, EXCH\$UTIL_FAO_BUFFER	
		50	DD	0009B	PUSHL	R0	
65		01	FB	0009D	CALLS	#1, LIB\$PUT_OUTPUT	
	F8	A3	DD	000A0	PUSHL	DEV	0314
7E		63	9A	000A3	MOVZBL	DEP, -(SP)	
7E	01	A3	9A	000A6	MOVZBL	DEP+1, -(SP)	
7E	02	A3	3C	000AA	MOVZWL	DEP+2, -(SP)	
	0000'	CF	9F	000AE	PUSHAB	P.AAT	
64		05	FB	000B2	CALLS	#5, EXCH\$UTIL_FAO_BUFFER	
		50	DD	000B5	PUSHL	R0	
65		01	FB	000B7	CALLS	#1, LIB\$PUT_OUTPUT	
	F8	A3	DD	000BA	PUSHL	DEVCHAR	0317
0000V	CF	01	FB	000BD	CALLS	#1, EXCH\$DBG_UTL_PRINT_DEVCHAR	
	50	01	D0	000C2	MOVL	#1, R0	0319
			04	000C5	RET		
		50	D4	000C6 2\$:	CLRL	R0	0320
			04	000C8	RET		

; Routine Size: 201 bytes, Routine Base: EXCH\$DBG\_CODE + 015C

```

229 0321 1 GLOBAL ROUTINE exch$dbg_fab_dump (fab : $ref_block) = %SBTTL 'exch$dbg_fab_dump'
230 0322 2 BEGIN
231 0323 2
232 0324 2 $dbgtrc_prefix ('fab_dump> ');
233 0325 2
234 0326 2 $print_fao (' Dump of fab at !XL:', .fab);
235 0327 2 ! $print_fao (' !15AS !XB - block id ', %ASCID 'FAB$B BID', .fab[FAB$B BID]);
236 0328 2 ! $print_fao (' !15AS !XB - block len ', %ASCID 'FAB$B BLN', .fab[FAB$B BLN]);
237 0329 2 ! $print_fao (' !15AS !XW - internal file index ', %ASCID 'FAB$W IFI', .fab[FAB$W IFI]);
238 0330 2 $print_fao (' !15AS !XL - file options ', %ASCID 'FAB$L FOP', .fab[FAB$L FOP]);
239 0331 2 $print_fao (' !15AS !XL - status ', %ASCID 'FAB$L STS', .fab[FAB$L STS]);
240 0332 2 $print_fao (' !15AS !XL - status value ', %ASCID 'FAB$L STV', .fab[FAB$L STV]);
241 0333 2 ! $print_fao (' !15AS !XL - allocation quantity ', %ASCID 'FAB$L ALQ', .fab[FAB$L ALQ]);
242 0334 2 ! $print_fao (' !15AS !XW - default allocation quantity ', %ASCID 'FAB$W DEQ', .fab[FA
243 0335 2 $print_fao (' !15AS !XB - file access ', %ASCID 'FAB$B FAC', .fab[FAB$B FAC]);
244 0336 2 $print_fao (' !15AS !XB - file sharing ', %ASCID 'FAB$B SHR', .fab[FAB$B SHR]);
245 0337 2 $print_fao (' !15AS !XL - user context ', %ASCID 'FAB$L CTX', .fab[FAB$L CTX]);
246 0338 2 $print_fao (' !15AS !XB - file organization ', %ASCID 'FAB$B ORG', .fab[FAB$B ORG]);
247 0339 2 $print_fao (' !15AS !XB - record format ', %ASCID 'FAB$B RAT', .fab[FAB$B RAT]);
248 0340 2 $print_fao (' !15AS !XB - record format ', %ASCID 'FAB$B RFM', .fab[FAB$B RFM]);
249 P 0341 2 $print_fao (' !15AS !XB !XL '!AF''', %ASCID 'filename',
250 0342 2 .fab[FAB$B FNS], .fab[FAB$L FNA], .fab[fab$b_fns], .fab[fab$l_fna]);
251 P 0343 2 $print_fao (' !15AS !XB !XL '!AF''', %ASCID 'default name',
252 0344 2 .fab[FAB$B DNS], .fab[FAB$L DNA], .fab[fab$b_dns], .fab[fab$l_dna]);
253 0345 2 $print_fao (' !15AS !XL - device characteristics ', %ASCID 'FAB$[DEV]', .fab[FAB$[DEV])
254 0346 2 exch$dbg_util_print_devchar (.fab[FAB$L_DEV]);
255 0347 2 $print_fao (' !15AS !XL - nam block address ', %ASCID 'FAB$L_NAM', .fab[FAB$L_NAM]);
256 0348 2 IF .fab[fab$l_nam] NEQ 0
257 0349 2 THEN exch$dbg_nam_dump (.fab[fab$l_nam]);
258 0350 2
259 0351 2 RETURN 1;
260 0352 1 END;

```

Line	Hex	Label	Text
62	002CC	P.AAW:	.ASCII \ Dump of fab at !XL:\<0>
61	002DB		
66	010E0017	P.AAV:	.LONG 17694743
20	00000000		.ADDRESS P.AAW
53	002EC	P.AAY:	.ASCII \ !15AS !XW - internal\
41	002FB		
35	0030A		
21	00314		.ASCII \ file index \<0><0><0>
20	00323		
20	010E0035	P.AAX:	.LONG 17694773
20	00000000		.ADDRESS P.AAY
00	0032C	P.ABA:	.ASCII \FAB\$W IFI\<0><0><0>
00	010E0009	P.AAZ:	.LONG 17694729
00	00000000		.ADDRESS P.ABA
53	00340	P.ABC:	.ASCII \ !15AS !XL - file options\
41	0034F		
35	0035E		
21	00368		.ASCII \ \<0><0>
20	010E002A	P.ABB:	.LONG 17694762
69	00000000		.ADDRESS P.ABC



```

20 20 53 41 35 31 21 20 20 20 20 20 20 20 00544 P.ACE: .ASCII \ !15AS !XB - record\
20 20 20 20 20 20 20 20 42 58 21 20 20 20 00553
00 00 64 72 6F 63 65 72 20 20 20 20 20 00562
00 00 00 20 20 74 61 6D 72 6F 66 20 0056C
010E0031 00578 P.ACD: .ASCII \ format \<0><0><0>
00000000' 0057C .LONG 17694769
00 00 00 54 41 52 5F 42 24 42 41 46 00580 P.ACG: .ADDRESS P.ACE
010E0009 0058C P.ACF: .ASCII \FAB$B RAT\<0><0><0>
00000000' 00590 .LONG 17694729
00594 P.ACI: .ADDRESS P.ACG
20 20 53 41 35 31 21 20 20 20 20 20 20 20 00594 P.ACI: .ASCII \ !15AS !XB - record\
20 20 20 20 20 20 20 20 42 58 21 20 20 20 005A3
00 00 64 72 6F 63 65 72 20 20 20 20 20 005B2
00 00 00 20 20 74 61 6D 72 6F 66 20 005BC
010E0031 005C8 P.ACH: .ASCII \ format \<0><0><0>
00000000' 005CC .LONG 17694769
00 00 00 4D 46 52 5F 42 24 42 41 46 005D0 P.ACK: .ADDRESS P.ACI
010E0009 005DC P.ACJ: .ASCII \FAB$B RFM\<0><0><0>
00000000' 005E0 .LONG 17694729
20 20 53 41 35 31 21 20 20 20 20 20 20 20 005E4 P.ACM: .ADDRESS P.ACK
41 21 22 20 20 20 20 4C 58 21 20 42 58 21 20 005F3
22 46 00602
010E0020 00604 P.ACL: .ASCII \ !15AS !XB !XL '!AF'\
00000000' 00608 .LONG 17694752
65 6D 61 6E 65 6C 69 66 0060C P.ACO: .ADDRESS P.ACM
010E0008 00614 P.ACN: .ASCII \filename\
00000000' 00618 .LONG 17694728
20 20 53 41 35 31 21 20 20 20 20 20 20 20 0061C P.ACQ: .ADDRESS P.ACO
41 21 22 20 20 20 20 4C 58 21 20 42 58 21 20 0062B
22 46 0063A
010E0020 0063C P.ACP: .ASCII \ !15AS !XB !XL '!AF'\
00000000' 00640 .LONG 17694752
65 6D 61 6E 20 74 6C 75 61 66 65 64 00644 P.ACS: .ADDRESS P.ACQ
010E000C 00650 P.ACR: .ASCII \default name\
00000000' 00654 .LONG 17694732
20 20 53 41 35 31 21 20 20 20 20 20 20 20 00658 P.ACU: .ADDRESS P.ACS
65 64 20 20 20 20 20 4C 58 21 20 20 20 20 00667
20 20 73 63 69 74 73 69 72 65 63 69 76 00676
010E0034 00680 P.ACT: .ASCII \cteristics \
00000000' 0068C .LONG 17694772
00 00 00 56 45 44 5F 4C 24 42 41 46 00690 P.ACW: .ADDRESS P.ACU
010E0009 006A0 P.ACV: .ASCII \FAB$L DEV\<0><0><0>
00000000' 006A4 .LONG 17694729
20 20 53 41 35 31 21 20 20 20 20 20 20 20 006AB P.ACY: .ADDRESS P.ACV
61 6E 20 20 20 20 20 4C 58 21 20 20 20 20 006B7
64 61 20 68 63 6F 6C 62 20 6D 006C6
00 20 20 73 73 65 72 64 006D0
010E002F 006D8 P.ACX: .ASCII \dress \<0>
00000000' 006DC .LONG 17694767
00 00 00 4D 41 4E 5F 4C 24 42 41 46 006E0 P.ADA: .ADDRESS P.ACY
010E0009 006EC P.ACZ: .ASCII \FAB$L NAM\<0><0><0>
00000000' 006F0 .LONG 17694729
006F0 .ADDRESS P.ADA

```

.PSECT EXCH\$DBG\_CODE,NOWRT,2



```

001C 00000 .ENTRY EXCH$DBG FAB_DUMP, Save R2,R3,R4
54 00000000G 00 9E 00002 MOVAB LIB$PUT_OUTPUT, R4 ; 0321
53 00000000G EF 9E 00009 MOVAB EXCH$UTIL_FAO_BUFFER, R3
52 04 AC D0 00010 MOVL FAB, R2 ; 0326
0000' 52 DD 00014 PUSHL R2
63 0000' CF 9F 00016 PUSHAB P.AAV
02 FB 0001A CALLS #2, EXCH$UTIL_FAO_BUFFER
50 DD 0001D PUSHL R0
64 01 FB 0001F CALLS #1, LIB$PUT_OUTPUT
7E 02 A2 3C 00022 MOVZWL 2(R2), -(SP) ; 0329
0000' CF 9F 00026 PUSHAB P.AAZ
0000' CF 9F 0002A PUSHAB P.AAX
63 03 FB 0002E CALLS #3, EXCH$UTIL_FAO_BUFFER
50 DD 00031 PUSHL R0
64 01 FB 00033 CALLS #1, LIB$PUT_OUTPUT
04 A2 DD 00036 PUSHL 4(R2) ; 0330
0000' CF 9F 00039 PUSHAB P.ABD
0000' CF 9F 0003D PUSHAB P.ABB
63 03 FB 00041 CALLS #3, EXCH$UTIL_FAO_BUFFER
50 DD 00044 PUSHL R0
64 01 FB 00046 CALLS #1, LIB$PUT_OUTPUT
08 A2 DD 00049 PUSHL 8(R2) ; 0331
0000' CF 9F 0004C PUSHAB P.ABH
0000' CF 9F 00050 PUSHAB P.ABF
63 03 FB 00054 CALLS #3, EXCH$UTIL_FAO_BUFFER
50 DD 00057 PUSHL R0
64 01 FB 00059 CALLS #1, LIB$PUT_OUTPUT
0C A2 DD 0005C PUSHL 12(R2) ; 0332
0000' CF 9F 0005F PUSHAB P.ABL
0000' CF 9F 00063 PUSHAB P.ABJ
63 03 FB 00067 CALLS #3, EXCH$UTIL_FAO_BUFFER
50 DD 0006A PUSHL R0
64 01 FB 0006C CALLS #1, LIB$PUT_OUTPUT
7E 16 A2 9A 0006F MOVZBL 22(R2), -(SP) ; 0335
0000' CF 9F 00073 PUSHAB P.ABP
0000' CF 9F 00077 PUSHAB P.ABN
63 03 FB 0007B CALLS #3, EXCH$UTIL_FAO_BUFFER
50 DD 0007E PUSHL R0
64 01 FB 00080 CALLS #1, LIB$PUT_OUTPUT
7E 17 A2 9A 00083 MOVZBL 23(R2), -(SP) ; 0336
0000' CF 9F 00087 PUSHAB P.ABT
0000' CF 9F 0008B PUSHAB P.ABR
63 03 FB 0008F CALLS #3, EXCH$UTIL_FAO_BUFFER
50 DD 00092 PUSHL R0
64 01 FB 00094 CALLS #1, LIB$PUT_OUTPUT
18 A2 DD 00097 PUSHL 24(R2) ; 0337
0000' CF 9F 0009A PUSHAB P.ABX
0000' CF 9F 0009E PUSHAB P.ABV
63 03 FB 000A2 CALLS #3, EXCH$UTIL_FAO_BUFFER
50 DD 000A5 PUSHL R0
64 01 FB 000A7 CALLS #1, LIB$PUT_OUTPUT
7E 1D A2 9A 000AA MOVZBL 29(R2), -(SP) ; 0338
0000' CF 9F 000AE PUSHAB P.ACB
0000' CF 9F 000B2 PUSHAB P.ABZ
63 03 FB 000B6 CALLS #3, EXCH$UTIL_FAO_BUFFER
50 DD 000B9 PUSHL R0
64 01 FB 000BB CALLS #1, LIB$PUT_OUTPUT

```

20  
41

67

20  
41

00

20  
64

22  
46

20  
65

72

7E	1E	A2	9A	000BE	MOVZBL	30(R2), -(SP)	:	0339
	0000'	CF	9F	000C2	PUSHAB	P.ACF	:	
	0000'	CF	9F	000C6	PUSHAB	P.ACD	:	
63		03	FB	000CA	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:	
		50	DD	000CD	PUSHL	R0	:	
64		01	FB	000CF	CALLS	#1, LIB\$PUT_OUTPUT	:	
7E	1F	A2	9A	000D2	MOVZBL	31(R2), -(SP)	:	0340
	0000'	CF	9F	000D6	PUSHAB	P.ACJ	:	
	0000'	CF	9F	000DA	PUSHAB	P.ACH	:	
63		03	FB	000DE	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:	
		50	DD	000E1	PUSHL	R0	:	
64		01	FB	000E3	CALLS	#1, LIB\$PUT_OUTPUT	:	
	2C	A2	DD	000E6	PUSHL	44(R2)	:	0342
7E	34	A2	9A	000E9	MOVZBL	52(R2), -(SP)	:	
	2C	A2	DD	000ED	PUSHL	44(R2)	:	
7E	34	A2	9A	000F0	MOVZBL	52(R2), -(SP)	:	
	0000'	CF	9F	000F4	PUSHAB	P.ACN	:	
	0000'	CF	9F	000F8	PUSHAB	P.ACL	:	
63		06	FB	000FC	CALLS	#6, EXCH\$UTIL_FAO_BUFFER	:	
		50	DD	000FF	PUSHL	R0	:	
64		01	FB	00101	CALLS	#1, LIB\$PUT_OUTPUT	:	
	30	A2	DD	00104	PUSHL	48(R2)	:	0344
7E	35	A2	9A	00107	MOVZBL	53(R2), -(SP)	:	
	30	A2	DD	0010B	PUSHL	48(R2)	:	
7E	35	A2	9A	0010E	MOVZBL	53(R2), -(SP)	:	
	0000'	CF	9F	00112	PUSHAB	P.ACR	:	
	0000'	CF	9F	00116	PUSHAB	P.ACP	:	
63		06	FB	0011A	CALLS	#6, EXCH\$UTIL_FAO_BUFFER	:	
		50	DD	0011D	PUSHL	R0	:	
64		01	FB	0011F	CALLS	#1, LIB\$PUT_OUTPUT	:	
	40	A2	DD	00122	PUSHL	64(R2)	:	0345
	0000'	CF	9F	00125	PUSHAB	P.ACV	:	
	0000'	CF	9F	00129	PUSHAB	P.ACT	:	
63		03	FB	0012D	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:	
		50	DD	00130	PUSHL	R0	:	
64		01	FB	00132	CALLS	#1, LIB\$PUT_OUTPUT	:	
	40	A2	DD	00135	PUSHL	64(R2)	:	0346
0000V	CF	01	FB	00138	CALLS	#1, EXCH\$DBG_UTL_PRINT_DEVCHAR	:	
	28	A2	DD	0013D	PUSHL	40(R2)	:	0347
	0000'	CF	9F	00140	PUSHAB	P.ACZ	:	
	0000'	CF	9F	00144	PUSHAB	P.ACX	:	
63		03	FB	00148	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:	
		50	DD	0014B	PUSHL	R0	:	
64		01	FB	0014D	CALLS	#1, LIB\$PUT_OUTPUT	:	
	28	A2	D5	00150	TSTL	40(R2)	:	0348
		08	13	00153	BEQL	1\$	:	
	28	A2	DD	00155	PUSHL	40(R2)	:	0349
0000V	CF	01	FB	00158	CALLS	#1, EXCH\$DBG_NAM_DUMP	:	
	50	01	D0	0015D	MOVL	#1, R0	:	0351
		04	00160	1\$:	RET		:	0352

; Routine Size: 353 bytes, Routine Base: EXCH\$DBG\_CODE + 0225

```

: 262 0353 1 GLOBAL ROUTINE exch$dbg_fab_name_print (fabb : $ref_bblock) = %SBTTL 'exch$dbg_fab_name_print'
: 263 0354 2 BEGIN
: 264 0355 2
: 265 0356 2 $dbgtrc_prefix ('fab_name_print> ');
: 266 0357 2
: 267 0358 2 LOCAL
: 268 0359 2     tmp_desc : $desc_block,           ! A descriptor for the total file name
: 269 0360 2     dev_desc : $desc_block,         ! A descriptor for the device name
: 270 0361 2     dir_desc : $desc_block,         ! A descriptor for the directory name
: 271 0362 2     dvi_desc : $desc_block,        ! A descriptor for the full device name
: 272 0363 2     nam_desc : $desc_block,         ! A descriptor for the name name
: 273 0364 2     nod_desc : $desc_block,        ! A descriptor for the node name
: 274 0365 2     typ_desc : $desc_block,        ! A descriptor for the type name
: 275 0366 2     ver_desc : $desc_block;         ! A descriptor for the version number
: 276 0367 2
: 277 0368 2 BIND
: 278 0369 2     nam = .fabb [fab$l_nam] : $bblock; ! Get pointer to the name block
: 279 0370 2
: 280 0371 2     tmp_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 281 0372 2     dev_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 282 0373 2     dir_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 283 0374 2     dvi_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 284 0375 2     nam_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 285 0376 2     nod_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 286 0377 2     typ_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 287 0378 2     ver_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 288 0379 2
: 289 0380 2 IF .fabb [fab$b_fns] NEQ 0
: 290 0381 2 THEN
: 291 0382 3     BEGIN
: 292 0383 3         tmp_desc [dsc$w_length] = .fabb [fab$b_fns]; ! Create file name desc
: 293 0384 3         tmp_desc [dsc$a_pointer] = .fabb [fab$l_fna]; ! ...
: 294 0385 3         $print_fao ('FAB filename ''!AS'', tmp_desc);
: 295 0386 3     END
: 296 0387 2 ELSE
: 297 0388 2     $print_lit ('FAB filename '' (null)');
: 298 0389 2
: 299 0390 2 IF NOT (.fabb [fab$l_sts])
: 300 0391 2 THEN
: 301 P 0392 2     $print_fao ('FAB$l_STS = !XL   FAB$l_STV = !XL',
: 302 0393 2         .fabb [fab$l_sts], .fabb [fab$l_stv]);
: 303 0394 2
: 304 0395 2 IF nam EQL 0
: 305 0396 2 THEN
: 306 0397 3     BEGIN
: 307 0398 3         $print_lit ('No nam block');
: 308 0399 3         RETURN -1;
: 309 0400 3     END;
: 310 0401 2
: 311 0402 2 IF .nam [nam$b_esl] GTRU 0
: 312 0403 2 THEN
: 313 0404 3     BEGIN
: 314 0405 3         tmp_desc [dsc$w_length] = .nam [nam$b_esl]; ! Create file name desc
: 315 0406 3         tmp_desc [dsc$a_pointer] = .nam [nam$l_esa]; ! ...
: 316 0407 3         $print_fao ('Expanded string ''!AS'', tmp_desc);
: 317 0408 3     END
: 318 0409 2 ELSE

```

```

: 319 0410 2   Sprint_lit ('Expanded string ''' (null)');
: 320 0411 2
: 321 0412 2   IF .nam [nam$b_rsl] GTRU 0
: 322 0413 2   THEN
: 323 0414 2   BEGIN
: 324 0415 2   tmp_desc [dsc$w_length] = .nam [nam$b_rsl];           ! Create file name desc
: 325 0416 2   tmp_desc [dsc$a_pointer] = .nam [nam$l_rsa];       ! ...
: 326 0417 2   Sprint_fao ('Result string ".AS"', tmp_desc);
: 327 0418 2   END;
: 328 0419 2
: 329 0420 2   dev_desc [dsc$w_length] = .nam [nam$b_dev];         ! Create file name desc
: 330 0421 2   dev_desc [dsc$a_pointer] = .nam [nam$l_dev];       ! ...
: 331 0422 2   dir_desc [dsc$w_length] = .nam [nam$b_dir];       ! Create file name desc
: 332 0423 2   dir_desc [dsc$a_pointer] = .nam [nam$l_dir];       ! ...
: 333 0424 2   BEGIN
: 334 0425 2   BIND
: 335 0426 2   len = nam [nam$t_dvi] : BYTE;
: 336 0427 2   dvi_desc [dsc$w_length] = .len;                   ! Create file name desc
: 337 0428 2   END;
: 338 0429 2   dvi_desc [dsc$a_pointer] = nam [nam$t_dvi]+1;     ! ...
: 339 0430 2   nam_desc [dsc$w_length] = .nam [nam$b_name];       ! Create file name desc
: 340 0431 2   nam_desc [dsc$a_pointer] = .nam [nam$l_name];     ! ...
: 341 0432 2   nod_desc [dsc$w_length] = .nam [nam$b_node];     ! Create file name desc
: 342 0433 2   nod_desc [dsc$a_pointer] = .nam [nam$l_node];     ! ...
: 343 0434 2   typ_desc [dsc$w_length] = .nam [nam$b_type];     ! Create file name desc
: 344 0435 2   typ_desc [dsc$a_pointer] = .nam [nam$l_type];     ! ...
: 345 0436 2   ver_desc [dsc$w_length] = .nam [nam$b_ver];     ! Create file name desc
: 346 0437 2   ver_desc [dsc$a_pointer] = .nam [nam$l_ver];     ! ...
: 347 0438 2
: 348 P 0439 2   Sprint_fao ('"!AS" "!AS" "!AS" "!AS" "!AS" "!AS" "!AS"
: 349 0440 2   nod_desc, dev_desc, dir_desc, nam_desc, typ_desc, ver_desc, dvi_desc);
: 350 0441 2
: 351 0442 2   RETURN 1;
: 352 0443 1   END;

```

														.PSECT EXCH\$DBG_PLIT, NOWRT, 2			
21	22	20	65	6D	61	6E	65	6C	69	66	20	42	41	46	006F4	P.ADC:	.ASCII \FAB filename '!AS'\<0><0>
										00	00	22	53	41	00703		
															010E0012	P.ADB:	.LONG 17694738
															00000000		.ADDRESS P.ADC
22	22	20	65	6D	61	6E	65	6C	69	66	20	42	41	46	00710	P.ADE:	.ASCII \FAB filename ''' (null)\<0><0>
						00	00	29	6C	6C	75	6E	28	20	0071F		
															010E0016	P.ADD:	.LONG 17694742
															00000000		.ADDRESS P.ADE
4C	58	21	20	3D	20	53	54	53	5F	4C	24	42	41	46	00730	P.ADG:	.ASCII \FAB\$l_STS = !XL FAB\$l_STV = !XL\<0>
3D	20	56	54	53	5F	4C	24	42	41	46	20	20	20	20	0073F		
										00	4C	58	21	20	0074E		
															00753		.ASCII <0>
															010E0022	P.ADF:	.LONG 17694754
															00000000		.ADDRESS P.ADG
			6B	63	6F	6C	62	20	6D	61	6E	20	6F	4E	0075C	P.ADI:	.ASCII \No nam block\
															010E000C	P.ADH:	.LONG 17694732
															00000000		.ADDRESS P.ADI
67	6E	69	72	74	73	20	64	65	64	6E	61	70	78	45	00770	P.ADK:	.ASCII \Expanded string '!AS'\<0><0><0>



		0B	A3	95	0009C	4\$:	TSTB	11(R3)	:	0402
			18	13	0009F		BEQL	5\$	:	
38	AE	0B	A3	9B	000A1		MOVZBW	11(R3), TMP_DESC	:	0405
3C	AE	0C	A3	00	000A6		MOVL	12(R3), TMP_DESC+4	:	0406
		38	AE	9F	000AB		PUSHAB	TMP_DESC	:	0407
		0000'	CF	9F	000AE		PUSHAB	P.ADJ	:	
		64		02	FB	000B2	CALLS	#2, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	000B5		PUSHL	R0	:	
			04	11	000B7		BRB	6\$	:	
		0000'	CF	9F	000B9	5\$:	PUSHAB	P.ADL	:	0410
		65		01	FB	000BD	CALLS	#1, LIB\$PUT_OUTPUT	:	
			03	A3	95	000C0	TSTB	3(R3)	:	0412
				19	13	000C3	BEQL	7\$	:	
38	AE	03	A3	9B	000C5		MOVZBW	3(R3), TMP_DESC	:	0415
3C	AE	04	A3	00	000CA		MOVL	4(R3), TMP_DESC+4	:	0416
		38	AE	9F	000CF		PUSHAB	TMP_DESC	:	0417
		0000'	CF	9F	000D2		PUSHAB	P.ADN	:	
		64		02	FB	000D6	CALLS	#2, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	000D9		PUSHL	R0	:	
		65		01	FB	000DB	CALLS	#1, LIB\$PUT_OUTPUT	:	
30	AE	39	A3	9B	000DE	7\$:	MOVZBW	57(R3), DEV_DESC	:	0420
34	AE	44	A3	00	000E3		MOVL	68(R3), DEV_DESC+4	:	0421
28	AE	3A	A3	9B	000E8		MOVZBW	58(R3), DIR_DESC	:	0422
2C	AE	48	A3	00	000ED		MOVL	72(R3), DIR_DESC+4	:	0423
20	AE	14	A3	9B	000F2		MOVZBW	20(R3), DVI_DESC	:	0427
24	AE	15	A3	9E	000F7		MOVAB	21(R3), DVI_DESC+4	:	0429
18	AE	3B	A3	9B	000FC		MOVZBW	59(R3), NAM_DESC	:	0430
1C	AE	4C	A3	00	00101		MOVL	76(R3), NAM_DESC+4	:	0431
10	AE	38	A3	9B	00106		MOVZBW	56(R3), NOD_DESC	:	0432
14	AE	40	A3	00	0010B		MOVL	64(R3), NOD_DESC+4	:	0433
08	AE	3C	A3	9B	00110		MOVZBW	60(R3), TYP_DESC	:	0434
0C	AE	50	A3	00	00115		MOVL	80(R3), TYP_DESC+4	:	0435
	6E	3D	A3	9B	0011A		MOVZBW	61(R3), VER_DESC	:	0436
04	AE	54	A3	00	0011E		MOVL	84(R3), VER_DESC+4	:	0437
		20	AE	9F	00123		PUSHAB	DVI_DESC	:	0440
		04	AE	9F	00126		PUSHAB	VER_DESC	:	
		10	AE	9F	00129		PUSHAB	TYP_DESC	:	
		24	AE	9F	0012C		PUSHAB	NAM_DESC	:	
		38	AE	9F	0012F		PUSHAB	DIR_DESC	:	
		44	AE	9F	00132		PUSHAB	DEV_DESC	:	
		28	AE	9F	00135		PUSHAB	NOD_DESC	:	
		0000'	CF	9F	00138		PUSHAB	P.ADP	:	
		64		08	FB	0013C	CALLS	#8, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	0013F		PUSHL	R0	:	
		65		01	FB	00141	CALLS	#1, LIB\$PUT_OUTPUT	:	0442
		50		01	00	00144	MOVL	#1, R0	:	0443
				04	00	00147	RET		:	

; Routine Size: 328 bytes, Routine Base: EXCH\$DBG\_CODE + 0386

```

354 0444 1 GLOBAL ROUTINE exch$dbg_dump_rt11_file (filb : $ref_bblock) = %SBTTL 'exch$dbg_dump_rt11_file'
355 0445 2 BEGIN
356 0446 2
357 0447 2 LOCAL
358 0448 2     buf : VECTOR [512, BYTE],
359 0449 2     pbn,
360 0450 2     cnt,
361 0451 2     status;
362 0452 2
363 0453 2 EXTERNAL ROUTINE
364 0454 2     exch$io_rt11_read;
365 0455 2
366 0456 2 BIND
367 0457 2     bufw = buf : VECTOR [256, WORD],
368 0458 2     ctx = filb [filb$a_context] : $ref_bblock,
369 0459 2     volb = filb [filb$a_assoc_volb] : $ref_bblock;
370 0460 2
371 0461 2 $dbgtrc_prefix ('dbg_dump_rt11_file> ');
372 0462 2
373 0463 2 $block_check (2, .filb, filb, 414);
374 0464 2 $block_check (2, .volb, volb, 415);
375 0465 2 $block_check (2, .ctx, rt11ctx, 416);
376 0466 2 $logic_check (2, (.ctx [rt11ctx$a_assoc_filb] EQL .filb), 106);
377 0467 2 $logic_check (2, (.ctx [rt11ctx$a_assoc_volb] EQL .volb), 107);
378 0468 2
379 0469 2 pbn = .ctx [rt11ctx$l_start_block];
380 0470 2 cnt = .ctx [rt11ctx$w_blocks];
381 0471 2
382 0472 2 INCRU blk FROM .pbn TO .pbn + .cnt - 1
383 0473 2 DO
384 0474 3     BEGIN
385 0475 3
386 0476 4     IF NOT (status = exch$io_rt11_read (.volb, .blk, 1, buf))
387 0477 3     THEN
388 0478 3         $exch_signal_return (.status);
389 0479 3
390 0480 3     $print_fao ('> !XW Vbn  (!XW Pbn)', .blk-.pbn, .blk);
391 0481 3
392 0482 3     INCRU i FROM 0 TO 15
393 0483 3     DO
394 0484 4         BEGIN
395 0485 4             REGISTER off, offw;
396 0486 4
397 0487 4             off = .i * 32;           offw = .i * 16;
398 0488 4
399 0489 4             $print_fao ('> !16(5XW) - !XW - '!AF''
400 P 0490 4                 .bufw [.offw + 15], .bufw [.offw + 14], .bufw [.offw + 13], .bufw [.offw + 12],
401 P 0491 4                 .bufw [.offw + 11], .bufw [.offw + 10], .bufw [.offw + 9], .bufw [.offw + 8],
402 P 0492 4                 .bufw [.offw + 7], .bufw [.offw + 6], .bufw [.offw + 5], .bufw [.offw + 4],
403 P 0493 4                 .bufw [.offw + 3], .bufw [.offw + 2], .bufw [.offw + 1], .bufw [.offw],
404 0494 4                 .off, 32, buf [.off]);
405 0495 4
406 0496 3         END;
407 0497 2     END;
408 0498 2
409 0499 2 RETURN 1;
410 0500 1 END;

```

69  
74

65  
65

69

69

```

.PSECT EXCH$DBG_PLIT,NOWRT,2
58 21 28 20 20 20 6E 62 56 20 57 58 21 20 3E 00808 P.ADS: .ASCII \> !XW Vbn (!XW Pbn)\<0><0><0>
00 00 00 29 6E 62 50 20 57 00817
010E0015 00820 P.ADR: .LONG 17694741
00000000' 00824 .ADDRESS P.ADS
2D 20 29 57 58 35 28 36 31 21 20 20 20 20 3E 00828 P.ADU: .ASCII \> !16(5XW) - !XW - '!AF'\<0>
00 22 46 41 21 22 20 2D 20 57 58 21 20 00837
010E001B 00844 P.ADT: .LONG 17694747
00000000' 00848 .ADDRESS P.ADU

.EXTRN EXCH$IO_RT11_READ
.EXTRN EXCH$_BADLOGIC

.PSECT EXCH$DBG_CODE,NOWRT,2
.OFFC 00000
.ENTRY EXCH$DBG_DUMP_RT11_FILE, Save R2,R3,R4,R5,- ; 0444
R6,R7,R8,R9,R10,R11
MOVAB EXCH$UTIL_FAO_BUFFER, R11
MOVAB LIB$STOP, R10
MOVL #EXCH$_BADLOGIC, R9
MOVAB EXCH$UTIL_BLOCK_CHECK, R8
MOVAB -512(SP), SP
ADDL3 #32, FILB, R3
ADDL3 #28, FILB, R6
MOVL #56295674, R2
MOVZWL #414, R1
MOVL FILB, R0
JSB EXCH$UTIL_BLOCK_CHECK
MOVL #68878579, R2
MOVZWL #415, R1
MOVL (R6), R0
JSB EXCH$UTIL_BLOCK_CHECK
MOVL (R3), R4
MOVL #8519924, R2
MOVZWL #416, R1
MOVL R4, R0
JSB EXCH$UTIL_BLOCK_CHECK
CMPL 16(R4), FILB
BEQL 1$
MOVZBL #106, -(SP)
PUSHL #1
PUSHL R9
CALLS #3, LIB$STOP
CMPL 20(R4), (R6)
BEQL 2$
MOVZBL #107, -(SP)
PUSHL #1
PUSHL R9
CALLS #3, LIB$STOP
MOVL 114(R4), PBN
MOVZWL 64(R4), CNT
MOVAB -1(CNT)[PBN], R5
MOVL PBN, BLK

```

69  
74

69  
74

67

62

69

6E  
20

20

72  
28

69

73

69  
76

61



		00B5	31	00097	BRW	6\$			
		5E	DD	0009A	3\$:	PUSHL	SP		0476
		01	DD	0009C		PUSHL	#1		
		52	DD	0009E		PUSHL	BLK		
		66	DD	000A0		PUSHL	(R6)		69
00000000G	EF	04	FB	000A2		CALLS	#4, EXCH\$IO_RT11_READ		
	57	50	DO	000A9		MOVL	R0, STATUS		
	10	57	EB	000AC		BLBS	STATUS, 4\$		69
	54	57	DO	000AF		MOVL	STATUS, TEMP	0478	
		54	DD	000B2		PUSHL	TEMP		
00000000G	00	01	FB	000B4		CALLS	#1, LIB\$SIGNAL		
	50	54	DO	000BB		MOVL	TEMP, R0		
		04	000BE			RET			6D
		52	DD	000BF	4\$:	PUSHL	BLK	0480	6F
7E	52	53	C3	000C1		SUBL3	PBN, BLK, -(SP)		
		CF	9F	000C5		PUSHAB	P.ADR		
	6B	03	FB	000C9		CALLS	#5, EXCH\$UTIL_FAO_BUFFER		
		50	DD	000CC		PUSHL	R0		
00000000G	00	01	FB	000CE		CALLS	#1, LIB\$PUT_OUTPUT		68
		54	D4	000D5		CLRL	I	0482	6E
50	54	05	78	000D7	5\$:	ASHL	#5, I, OFF	0487	
51	54	04	78	000DB		ASHL	#4, I, OFFW		
		6E40	9F	000DF		PUSHAB	BUF[OFF]	0494	
		20	DD	000E2		PUSHL	#32		
		50	DD	000E4		PUSHL	OFF		69
7E	0C	AE41	3C	000E6		MOVZWL	BUFW[OFFW], -(SP)		
7E	12	AE41	3C	000EB		MOVZWL	BUFW+2[OFFW], -(SP)		
7E	18	AE41	3C	000F0		MOVZWL	BUFW+4[OFFW], -(SP)		69
7E	1E	AE41	3C	000F5		MOVZWL	BUFW+6[OFFW], -(SP)		69
7E	24	AE41	3C	000FA		MOVZWL	BUFW+8[OFFW], -(SP)		
7E	2A	AE41	3C	000FF		MOVZWL	BUFW+10[OFFW], -(SP)		
7E	30	AE41	3C	00104		MOVZWL	BUFW+12[OFFW], -(SP)		73
7E	36	AE41	3C	00109		MOVZWL	BUFW+14[OFFW], -(SP)		
7E	3C	AE41	3C	0010E		MOVZWL	BUFW+16[OFFW], -(SP)		
7E	42	AE41	3C	00113		MOVZWL	BUFW+18[OFFW], -(SP)		69
7E	48	AE41	3C	00118		MOVZWL	BUFW+20[OFFW], -(SP)		74
7E	4E	AE41	3C	0011D		MOVZWL	BUFW+22[OFFW], -(SP)		
7E	54	AE41	3C	00122		MOVZWL	BUFW+24[OFFW], -(SP)		
7E	5A	AE41	3C	00127		MOVZWL	BUFW+26[OFFW], -(SP)		
7E	60	AE41	3C	0012C		MOVZWL	BUFW+28[OFFW], -(SP)		
7E	66	AE41	3C	00131		MOVZWL	BUFW+30[OFFW], -(SP)		
		0000	CF	9F	00136	PUSHAB	P.ADR		63
	6B	14	FB	0013A		CALLS	#20, EXCH\$UTIL_FAO_BUFFER		69
		50	DD	0013D		PUSHL	R0		
00000000G	00	01	FB	0013F		CALLS	#1, LIB\$PUT_OUTPUT		
		54	D6	00146		INCL	I	0482	
	0F	54	D1	00148		CMPL	I, #15		63
		8A	1B	0014B		BLEQU	5\$		69
		52	D6	0014D		INCL	BLK	0472	
	55	52	D1	0014F	6\$:	CMPL	BLK, R5		
		03	1A	00152		BGTRU	7\$		
		FF43	31	00154		BRW	3\$		
	50	01	DO	00157	7\$:	MOVL	#1, R0	0499	61
		04	0015A			RET		0500	63

; Routine Size: 347 bytes, Routine Base: EXCH\$DBG\_CODE + 04CE

EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
exch\$dbg\_dump\_rtl1\_file

K 6  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 24  
(7)

EXCI  
V04.

69  
6E

68  
6E

68  
69

```

412 0501 1 GLOBAL ROUTINE exch$dbg_nam_dump (nam : $ref_bblock) = %SBTTL 'exch$dbg_nam_dump'
413 0502 2 BEGIN
414 0503 2
415 0504 2 $dbgtrc_prefix ('nam_dump> ');
416 0505 2
417 0506 2 |
418 0507 2 | A macro to add successive ASCII items to a string buffer, and update the
419 0508 2 | descriptor for the string.
420 0509 2 |
421 0510 2 MACRO
422 0511 2     add_to_list (string) =
423 0512 2     BEGIN
424 0513 2         BIND desc = %ASCID string : $BBLOCK;
425 0514 2         CH$MOVE(.desc[dsc$w_length]
426 0515 2             .desc[dsc$a_pointer],
427 0516 2             .ptr[1] + .ptr[0]);
428 0517 2         ptr[0] = .ptr[0] + .desc[dsc$w_length];
429 0518 2     END%;
430 0519 2
431 0520 2 LOCAL
432 0521 2     buffer : VECTOR [255, BYTE],
433 0522 2     ptr : VECTOR [2, LONG];
434 0523 2
435 0524 2     $print_fao ('      Dump of nam at !XL:', .nam);
436 0525 2     $print_fao ('          !15AS      !XB      - block id ', %ASCID 'NAM$B BID', .nam[NAM$B BID]);
437 0526 2     $print_fao ('          !15AS      !XB      - block length ', %ASCID 'NAM$B BLN', .nam[NAM$B BLN]);
438 0527 2     $print_fao ('          !15AS      !XB      - resultant string area size ', %ASCID 'NAM$B RSS', .nam[
439 0528 2     $print_fao ('          !15AS      !XB      - name options ', %ASCID 'NAM$B NOP', .nam[NAM$B NOP]);
440 0529 2     $print_fao ('          !15AS      !XL      - wild card context ', %ASCID 'NAM$SL_WCC', .nam[NAM$SL_WCC]);
441 0530 2     ptr [0] = 0;
442 0531 2     ptr [1] = buffer;
443 0532 2     IF .nam[nam$l_fnb] NEQ 0
444 0533 2     THEN
445 0534 2     BEGIN
446 0535 2         IF .nam[NAM$V_EXP_VER] THEN add_to_list('xver ');
447 0536 2         IF .nam[NAM$V_EXP_TYPE] THEN add_to_list('xtyp ');
448 0537 2         IF .nam[NAM$V_EXP_NAME] THEN add_to_list('xnam ');
449 0538 2         IF .nam[NAM$V_WILD_VER] THEN add_to_list('*ver ');
450 0539 2         IF .nam[NAM$V_WILD_TYPE] THEN add_to_list('*typ ');
451 0540 2         IF .nam[NAM$V_WILD_NAME] THEN add_to_list('*nam ');
452 0541 2         IF .nam[NAM$V_EXP_DIR] THEN add_to_list('xdir ');
453 0542 2         IF .nam[NAM$V_EXP_DEV] THEN add_to_list('xdev ');
454 0543 2         IF .nam[NAM$V_WILD_CARD] THEN add_to_list('*** ');
455 0544 2         IF .nam[NAM$V_SEARCH_LIST] THEN add_to_list('schl ');
456 0545 2         IF .nam[NAM$V_CNCL_DEV] THEN add_to_list('cncl ');
457 0546 2         IF .nam[NAM$V_ROOT_DIR] THEN add_to_list('root ');
458 0547 2         IF .nam[NAM$V_LOWVER] THEN add_to_list('lowv ');
459 0548 2         IF .nam[NAM$V_HIGHVER] THEN add_to_list('hiv ');
460 0549 2         IF .nam[NAM$V_PPF] THEN add_to_list('ppf ');
461 0550 2         IF .nam[NAM$V_NODE] THEN add_to_list('nod ');
462 0551 2         IF .nam[NAM$V_QUOTED] THEN add_to_list('quot ');
463 0552 2         IF .nam[NAM$V_WILD_DIR] THEN add_to_list('*dir ');
464 0553 2         IF .nam[NAM$V_GRP_MBR] THEN BEGIN
465 0554 2             add_to_list('grpmem ');
466 0555 2             IF .nam[NAM$V_WILD_GRP] THEN add_to_list('*grp ');
467 0556 2             IF .nam[NAM$V_WILD_MBR] THEN add_to_list('*mbr ');
468 0557 2         END;

```

```

: 469      0558      3      IF .ptr[0] GTR 0 THEN ptr[0] = .ptr[0] - 1;      ! Remove final space
: 470      0559      2      END;
: 471      0560      2      $print_fao ('      !15AS      !XL      - name bits '!AS'', %ASCID 'NAM$L_FNB', .nam[NAM$L_FNB], ptr);
: 472      P 0561      2      $print_fao ('      !15AS      !XB !XL      '!AF'', %ASCID 'expanded string',
: 473      0562      2      .nam[NAM$B_ESL], .nam[NAM$L_ESA], .nam[nam$b_esl], .nam[nam$l_esa]);
: 474      P 0563      2      $print_fao ('      !15AS      !XB !XL      '!AF'', %ASCID 'result string',
: 475      0564      2      .nam[NAM$B_RSL], .nam[NAM$L_RSA], .nam[nam$b_rs], .nam[nam$l_rsa]);
: 476      0565      2      $print_fao ('      !15AS      '!AC'' - device name, %ASCID 'NAM$T_DVI', nam[NAM$T_DVI]);
: 477      P 0566      2      $print_fao ('      '!AF'' '!AF'' '!AF'' '!AF'' '!AF'' '!AF''
: 478      P P 0567      2      .nam[NAM$B_NODE], .nam[NAM$L_NODE],
: 479      P P 0568      2      .nam[NAM$B_DEV], .nam[NAM$L_DEV],
: 480      P P 0569      2      .nam[NAM$B_DIR], .nam[NAM$L_DIR],
: 481      P P 0570      2      .nam[NAM$B_NAME], .nam[NAM$L_NAME],
: 482      P 0571      2      .nam[NAM$B_TYPE], .nam[NAM$L_TYPE],
: 483      0572      2      .nam[NAM$B_VER], .nam[NAM$L_VER]);
: 484      0573      2      $print_fao ('      !15AS      !XL      - related file nam block addr ', %ASCID 'NAM$L_RLF', .nam[NAM$L_
: 485      0574      2      IF .nam[NAM$L_RLF] NEQ 0
: 486      0575      2      THEN exch$dbg_nam_dump (.nam[NAM$L_RLF]);
: 487      0576      2
: 488      0577      2 RETURN 1;
: 489      0578      1 END;

```

														.PSECT EXCH\$DBG_PLIT,NOWRT,2				
6D	61	6E	20	66	6F	20	70	6D	75	44	20	20	20	20	0084C	P.ADW:	.ASCII \	Dump of nam at !XL:\<0>
						00	3A	4C	58	21	20	74	61	20	0085B			
													010E0017	00864	P.ADV:	.LONG	17694743	
													00000000	00868		.ADDRESS	P.ADW	
20	20	53	41	35	31	21	20	20	20	20	20	20	20	20	0086C	P.ADY:	.ASCII \	!15AS !XB - name o\
20	20	20	20	20	20	20	20	42	58	21	20	20	20	20	0087B			
					6F	20	65	6D	61	6E	20	2D	20	20	0088A			
							00	20	73	6E	6F	69	74	70	00894		.ASCII \options \<0>	
													010E002F	0089C	P.ADX:	.LONG	17694767	
													00000000	008A0		.ADDRESS	P.ADY	
			00	00	00	50	4F	4E	5F	42	24	4D	41	4E	008A4	P.AEA:	.ASCII \NAM\$B_NOP\<0><0><0>	
													010E0009	008B0	P.ADZ:	.LONG	17694729	
													00000000	008B4		.ADDRESS	P.AEA	
20	20	53	41	35	31	21	20	20	20	20	20	20	20	20	008B8	P.AEC:	.ASCII \	!15AS !XL - wild card co\
69	77	20	20	20	20	20	20	4C	58	21	20	20	20	20	008C7			
					6F	63	20	64	72	61	63	20	64	6C	008D6			
							00	20	20	74	78	65	74	6E	008E0		.ASCII \ntext \<0>	
													010E002F	008E8	P.AEB:	.LONG	17694767	
													00000000	008EC		.ADDRESS	P.AEC	
			00	00	00	43	43	57	5F	4C	24	4D	41	4E	008F0	P.AEE:	.ASCII \NAM\$L_WCC\<0><0><0>	
													010E0009	008FC	P.AED:	.LONG	17694729	
													00000000	00900		.ADDRESS	P.AEE	
						00	00	00	20	72	65	76	78	00904	P.AEG:	.ASCII \xver \<0><0><0>		
													010E0005	0090C	P.AEF:	.LONG	17694725	
													00000000	00910		.ADDRESS	P.AEG	
						00	00	00	20	70	79	74	78	00914	P.AEI:	.ASCII \xtyp \<0><0><0>		
													010E0005	0091C	P.AEH:	.LONG	17694725	
													00000000	00920		.ADDRESS	P.AEI	
			00	00	00	20	6D	61	6E	78	00924	P.AEK:	.ASCII \xnam \<0><0><0>					
													010E0005	0092C	P.AEJ:	.LONG	17694725	
													00000000	00930		.ADDRESS	P.AEK	

```

00 00 00 20 72 65 76 2A 00934 P.AEM: .ASCII \*ver \<0><0><0>
010E0005 0093C P.AEL: .LONG 17694725
00000000 00940 .ADDRESS P.AEM
00 00 00 20 70 79 74 2A 00944 P.AEO: .ASCII \*typ \<0><0><0>
010E0005 0094C P.AEN: .LONG 17694725
00000000 00950 .ADDRESS P.AEO
00 00 00 20 6D 61 6E 2A 00954 P.AEQ: .ASCII \*nam \<0><0><0>
010E0005 0095C P.AEP: .LONG 17694725
00000000 00960 .ADDRESS P.AEQ
00 00 00 20 72 69 64 78 00964 P.AES: .ASCII \xdir \<0><0><0>
010E0005 0096C P.AER: .LONG 17694725
00000000 00970 .ADDRESS P.AES
00 00 00 20 76 65 64 78 00974 P.AEU: .ASCII \xdev \<0><0><v>
010E0005 0097C P.AET: .LONG 17694725
00000000 00980 .ADDRESS P.AEU
20 2A 2A 2A 00984 P.AEW: .ASCII \*** \
010E0004 00988 P.AEV: .LONG 17694724
00000000 0098C .ADDRESS P.AEW
00 00 00 20 6C 68 63 73 00990 P.AEY: .ASCII \schl \<0><0><0>
010E0005 00998 P.AEX: .LONG 17694725
00000000 0099C .ADDRESS P.AEY
00 00 00 20 6C 63 6E 63 009A0 P.AFA: .ASCII \cncl \<0><0><0>
010E0005 009A8 P.AEZ: .LONG 17694725
00000000 009AC .ADDRESS P.AFA
00 00 00 20 74 6F 6F 72 009B0 P.AFC: .ASCII \root \<0><0><0>
010E0005 009B8 P.AFB: .LONG 17694725
00000000 009BC .ADDRESS P.AFC
00 00 00 20 76 77 6F 6C 009C0 P.AFE: .ASCII \lowv \<0><0><0>
010E0005 009C8 P.AFD: .LONG 17694725
00000000 009CC .ADDRESS P.AFE
20 76 69 68 009D0 P.AFG: .ASCII \hiv \
010E0004 009D4 P.AFF: .LONG 17694724
00000000 009D8 .ADDRESS P.AFG
20 66 70 70 009DC P.AFI: .ASCII \ppf \
010E0004 009E0 P.AFH: .LONG 17694724
00000000 009E4 .ADDRESS P.AFI
20 64 6F 6E 009E8 P.AFK: .ASCII \nod \
010E0004 009EC P.AFJ: .LONG 17694724
00000000 009F0 .ADDRESS P.AFK
00 00 00 20 74 6F 75 71 009F4 P.AFM: .ASCII \quot \<0><0><0>
010E0005 009FC P.AFL: .LONG 17694725
00000000 00A00 .ADDRESS P.AFM
00 00 00 20 72 69 64 2A 00A04 P.AFO: .ASCII \*dir \<0><0><0>
010E0005 00A0C P.AFN: .LONG 17694725
00000000 00A10 .ADDRESS P.AFO
00 20 6D 65 6D 70 72 67 00A14 P.AFQ: .ASCII \grpmem \<0>
010E0007 00A1C P.AFP: .LONG 17694727
00000000 00A20 .ADDRESS P.AFQ
00 00 00 20 70 72 67 2A 00A24 P.AFS: .ASCII \*grp \<0><0><0>
010E0005 00A2C P.AFR: .LONG 17694725
00000000 00A30 .ADDRESS P.AFS
00 00 00 20 72 62 6D 2A 00A34 P.AFU: .ASCII \*mbr \<0><0><0>
010E0005 00A3C P.AFT: .LONG 17694725
00000000 00A40 .ADDRESS P.AFU
20 20 53 41 35 31 21 20 20 20 20 20 20 20 20 00A44 P.AFW: .ASCII \ !15AS !XL - name bits '! \
61 6E 20 2D 20 20 20 20 4C 58 21 20 20 20 20 00A53
21 22 20 73 74 69 62 20 65 6D 00A62

```

S  
R  
E  
L  
M  
C









50	04	AE		15	18	0018B	BGEQ	15\$		
60	00CC	D8	00C8	6E	C1	0018D	ADDL3	PTR, PTR+4, R0		
		50	00C8	C8	28	00192	MOVC3	DESC, @DESC+4, (R0)		
		6E		C8	3C	0019A	MOVZWL	DESC, R0		
		15	02	50	C0	0019F	ADDL2	R0, PTR		
50	04	AE		A7	E9	001A2	15\$:	BLBC	2(R7), 16\$	0549
60	00D8	D8	00D4	6E	C1	001A6	ADDL3	PTR, PTR+4, R0		
		50	00D4	C8	28	001AB	MOVC3	DESC, @DESC+4, (R0)		
		6E		C8	3C	001B3	MOVZWL	DESC, R0		
15		67		50	C0	001B8	ADDL2	R0, PTR		
50	04	AE		11	E1	001BB	16\$:	BBC	#17, (R7), 17\$	0550
60	00E4	D8	00E0	6E	C1	001BF	ADDL3	PTR, PTR+4, R0		
		50	00E0	C8	28	001C4	MOVC3	DESC, @DESC+4, (R0)		
		6E		C8	3C	001CC	MOVZWL	DESC, R0		
15		67		50	C0	001D1	ADDL2	R0, PTR		
50	04	AE		12	E1	001D4	17\$:	BBC	#18, (R7), 18\$	0551
60	00F4	D8	00F0	6E	C1	001D8	ADDL3	PTR, PTR+4, R0		
		50	00F0	C8	28	001DD	MOVC3	DESC, @DESC+4, (R0)		
		6E		C8	3C	001E5	MOVZWL	DESC, R0		
15		67		50	C0	001EA	ADDL2	R0, PTR		
50	04	AE		14	E1	001ED	18\$:	BBC	#20, (R7), 19\$	0552
60	0104	D8	0100	6E	C1	001F1	ADDL3	PTR, PTR+4, R0		
		50	0100	C8	28	001F6	MOVC3	DESC, @DESC+4, (R0)		
		6E		C8	3C	001FE	MOVZWL	DESC, R0		
47		67		50	C0	00203	ADDL2	R0, PTR		
50	04	AE		13	E1	00206	19\$:	BBC	#19, (R7), 21\$	0553
60	0114	D8	0110	6E	C1	0020A	ADDL3	PTR, PTR+4, R0		0554
		50	0110	C8	28	0020F	MOVC3	DESC, @DESC+4, (R0)		
		6E		C8	3C	00217	MOVZWL	DESC, R0		
		15	03	50	C0	0021C	ADDL2	R0, PTR		
50	04	AE		A7	E9	0021F	BLBC	3(R7), 20\$		0555
60	0124	D8	0120	6E	C1	00223	ADDL3	PTR, PTR+4, R0		
		50	0120	C8	28	00228	MOVC3	DESC, @DESC+4, (R0)		
		6E		C8	3C	00230	MOVZWL	DESC, R0		
15		67		50	C0	00235	ADDL2	R0, PTR		
50	04	AE		19	E1	00238	20\$:	BBC	#25, (R7), 21\$	0556
60	0134	D8	0130	6E	C1	0023C	ADDL3	PTR, PTR+4, R0		
		50	0130	C8	28	00241	MOVC3	DESC, @DESC+4, (RC)		
		6E		C8	3C	00249	MOVZWL	DESC, R0		
				50	C0	0024E	ADDL2	R0, PTR		
				6E	D5	00251	21\$:	TSTL	PTR	0558
				02	15	00253	BLEQ	22\$		
				6E	D7	00255	DECL	PTR		
				5E	DD	00257	22\$:	PUSHL	SP	0560
				67	DD	00259	PUSHL	(R7)		
			0178	C8	9F	0025B	PUSHAB	P.AFX		
			0164	C8	9F	0025F	PUSHAB	P.AFV		
		69		04	FB	00263	CALLS	#4, EXCH\$UTIL_FAO_BUFFER		
		6A		50	DD	00266	PUSHL	R0		
				01	FB	00268	CALLS	#1, LIB\$PUT_OUTPUT		
		7E	0C	A6	DD	0026B	PUSHL	12(R6)		0562
			0B	A6	9A	0026E	MOVZBL	11(R6), -(SP)		
		7E	0C	A6	DD	00272	PUSHL	12(R6)		
			0B	A6	9A	00275	MOVZBL	11(R6), -(SP)		
			01B8	C8	9F	00279	PUSHAB	P.AGB		
			01A0	C8	9F	0027D	PUSHAB	P.AFZ		
		69		06	FB	00281	CALLS	#6, EXCH\$UTIL_FAO_BUFFER		

6A		50	DD	00284	PUSHL	R0		
		01	FB	00286	CALLS	#1, LIB\$PUT_OUTPUT		
	04	A6	DD	00289	PUSHL	4(R6)		0564
7E	03	A6	9A	0028C	MOVZBL	3(R6), -(SP)		
	04	A6	DD	00290	PUSHL	4(R6)		
7E	03	A6	9A	00293	MOVZBL	3(R6), -(SP)		
	01F8	C8	9F	00297	PUSHAB	P.AGF		
	01E0	C8	9F	0029B	PUSHAB	P.AGD		
69		06	FB	0029F	CALLS	#6, EXCH\$UTIL_FAO_BUFFER		
		50	DD	002A2	PUSHL	R0		
6A		01	FB	002A4	CALLS	#1, LIB\$PUT_OUTPUT		
	14	A6	9F	002A7	PUSHAB	20(R6)		0565
	023C	C8	9F	002AA	PUSHAB	P.AGJ		
	0228	C8	9F	002AE	PUSHAB	P.AGH		
69		03	FB	002B2	CALLS	#3, EXCH\$UTIL_FAO_BUFFER		
		50	DD	002B5	PUSHL	R0		
6A		01	FB	002B7	CALLS	#1, LIB\$PUT_OUTPUT		
	54	A6	DD	002BA	PUSHL	84(R6)		0572
7E	3D	A6	9A	002BD	MOVZBL	61(R6), -(SP)		
	50	A6	DD	002C1	PUSHL	80(R6)		
7E	3C	A6	9A	002C4	MOVZBL	60(R6), -(SP)		
	4C	A6	DD	002C8	PUSHL	76(R6)		
7E	3B	A6	9A	002CB	MOVZBL	59(R6), -(SP)		
	48	A6	DD	002CF	PUSHL	72(R6)		
7E	3A	A6	9A	002D2	MOVZBL	58(R6), -(SP)		
	44	A6	DD	002D6	PUSHL	68(R6)		
7E	39	A6	9A	002D9	MOVZBL	57(R6), -(SP)		
	40	A6	DD	002DD	PUSHL	64(R6)		
7E	38	A6	9A	002E0	MOVZBL	56(R6), -(SP)		
	0270	C8	9F	002E4	PUSHAB	P.AGL		
69		0D	FB	002E8	CALLS	#13, EXCH\$UTIL_FAO_BUFFER		
		50	DD	002EB	PUSHL	R0		
6A		01	FB	002ED	CALLS	#1, LIB\$PUT_OUTPUT		
	10	A6	DD	002F0	PUSHL	16(R6)		0573
	02C8	C8	9F	002F3	PUSHAB	P.AGP		
	02B4	C8	9F	002F7	PUSHAB	P.AGN		
69		03	FB	002FB	CALLS	#3, EXCH\$UTIL_FAO_BUFFER		
		50	DD	002FE	PUSHL	R0		
6A		01	FB	00300	CALLS	#1, LIB\$PUT_OUTPUT		
	10	A6	D5	00303	TSTL	16(R6)		0574
		08	13	00306	BEQL	23\$		
	10	A6	DD	00308	PUSHL	16(R6)		0575
FCF0	CF	01	FB	0030B	CALLS	#1, EXCH\$DBG_NAM_DUMP		
	50	01	DD	00310	MOVL	#1, R0		0577
		04	00313	23\$:	RET			0578

; Routine Size: 788 bytes, Routine Base: EXCH\$DBG\_CODE + 0629

```
491 0579 1 GLOBAL ROUTINE exch$dbg_utl_print_devchar (dev : $bblock) = %SBTTL 'exch$dbg_utl_print_devchar'
492 0580 2 BEGIN
493 0581 2
494 0582 2 $dbgtrc_prefix ('dbg_utl_print_devchar> ');
495 0583 2 $trace_print_fao ('characteristics longword .XL', .dev [0,0,32,0]);
496 0584 2
497 0585 2 IF .dev [dev$V_rec]
498 0586 2 THEN
499 0587 2 $print_lit (' rec - device is record oriented');
500 0588 2 IF .dev [dev$V_ccl]
501 0589 2 THEN
502 0590 2 $print_lit (' ccl - carriage control device');
503 0591 2 IF .dev [dev$V_trm]
504 0592 2 THEN
505 0593 2 $print_lit (' trm - device is a terminal');
506 0594 2 IF .dev [dev$V_dir]
507 0595 2 THEN
508 0596 2 $print_lit (' dir - device is directory structured');
509 0597 2 IF .dev [dev$V_sdi]
510 0598 2 THEN
511 0599 2 $print_lit (' sdi - device is single directory structured');
512 0600 2 IF .dev [dev$V_sqd]
513 0601 2 THEN
514 0602 2 $print_lit (' sqd - sequential block-oriented device (i.e., magtape)');
515 0603 2 IF .dev [dev$V_spl]
516 0604 2 THEN
517 0605 2 $print_lit (' spl - device being spooled');
518 0606 2 IF .dev [dev$V_opr]
519 0607 2 THEN
520 0608 2 $print_lit (' opr - device is an operator');
521 0609 2 IF .dev [dev$V_rct]
522 0610 2 THEN
523 0611 2 $print_lit (' rct - disk contains rct (dec standard 166 disk)');
524 0612 2 IF .dev [dev$V_net]
525 0613 2 THEN
526 0614 2 $print_lit (' net - network device');
527 0615 2 IF .dev [dev$V_fod]
528 0616 2 THEN
529 0617 2 $print_lit (' fod - files-oriented device (i.e., disk and mt)');
530 0618 2 IF .dev [dev$V_dua]
531 0619 2 THEN
532 0620 2 $print_lit (' dua - device is dual ported');
533 0621 2 IF .dev [dev$V_shr]
534 0622 2 THEN
535 0623 2 $print_lit (' shr - device shareable');
536 0624 2 IF .dev [dev$V_gen]
537 0625 2 THEN
538 0626 2 $print_lit (' gen - device is a generic device');
539 0627 2 IF .dev [dev$V_avl]
540 0628 2 THEN
541 0629 2 $print_lit (' avl - device available for use');
542 0630 2 IF .dev [dev$V_mnt]
543 0631 2 THEN
544 0632 2 $print_lit (' mnt - device is mounted');
545 0633 2 IF .dev [dev$V_mbx]
546 0634 2 THEN
547 0635 2 $print_lit (' mbx - device is a mailbox');
```

```

548 0636 2 IF .dev [dev$V_dmt]
549 0637 2 THEN
550 0638 2   $print_lit (' dmt - device marked for dismount');
551 0639 2 IF .dev [dev$V_elg]
552 0640 2 THEN
553 0641 2   $print_lit (' elg - device has error logging enabled');
554 0642 2 IF .dev [dev$V_all]
555 0643 2 THEN
556 0644 2   $print_lit (' all - device is allocated');
557 0645 2 IF .dev [dev$V_for]
558 0646 2 THEN
559 0647 2   $print_lit (' for - device is mounted foreign (i.e., non-file structured)');
560 0648 2 IF .dev [dev$V_swl]
561 0649 2 THEN
562 0650 2   $print_lit (' swl - device is software write locked');
563 0651 2 IF .dev [dev$V_idv]
564 0652 2 THEN
565 0653 2   $print_lit (' idv - device capable of providing input');
566 0654 2 IF .dev [dev$V_odv]
567 0655 2 THEN
568 0656 2   $print_lit (' odv - device capable of providing output');
569 0657 2 IF .dev [dev$V_rnd]
570 0658 2 THEN
571 0659 2   $print_lit (' rnd - device allows random access');
572 0660 2 IF .dev [dev$V_rtm]
573 0661 2 THEN
574 0662 2   $print_lit (' rtm - device is realtime in nature');
575 0663 2 IF .dev [dev$V_rck]
576 0664 2 THEN
577 0665 2   $print_lit (' rck - device has read checking enabled');
578 0666 2 IF .dev [dev$V_wck]
579 0667 2 THEN
580 0668 2   $print_lit (' wck - device has write checking enabled');
581 0669 2
582 0670 2 RETURN 1;
583 0671 1 END;

```

.PSECT EXCH\$DBG\_PLIT,NOWRT,2

```

69 20 65 63 69 76 65 64 20 2D 20 63 65 72 09 00BDC P.AGS: .ASCII <9>\rec - device is record oriented\
74 6E 65 69 72 6F 20 64 72 6F 63 65 72 20 73 00BEB
64 65 00BFA
010E0020 00BFC P.AGR: .LONG 17694752
00000000' 00C00 .ADDRESS P.AGS
65 67 61 69 72 72 61 63 20 2D 20 6C 63 63 09 00C04 P.AGU: .ASCII <9>\ccl - carriage control device\<0>
65 63 69 76 65 64 20 6C 6F 72 74 6E 6F 63 20 00C13
00 00C22
00 00C23
010E001E 00C24 P.AGT: .ASCII <0>
00000000' 00C28 .LONG 17694750
 .ADDRESS P.AGU
69 20 65 63 69 76 65 64 20 2D 20 6D 72 74 09 00C2C P.AGW: .ASCII <9>\trm - device is a terminal\<0>
00 6C 61 6E 69 6D 72 65 74 20 61 20 73 00C3B
010E001B 00C48 P.AGV: .LONG 17694747
00000000' 00C4C .ADDRESS P.AGW
69 20 65 63 69 76 65 64 20 2D 20 72 69 64 09 00C50 P.AGY: .ASCII <9>\dir - device is directory structur\

```



```

73 75 20 72 6F 66 20 65 6C 62 61 6C 69 61 76 00E4B
        00 65 00E5A
        010E001F 00E5C P.AHT: .LONG 17694751
        00000000 00E60 .ADDRESS P.AHU
69 20 65 63 69 76 65 64 20 2D 20 74 6E 6D 09 00E64 P.AHW: .ASCII <9>\mnt - device is mounted\
        64 65 74 6E 75 6F 6D 20 73 00E73
        010E0018 00E7C P.AHV: .LONG 17694744
        00000000 00E80 .ADDRESS P.AHW
69 20 65 63 69 76 65 64 20 2D 20 78 62 6D 09 00E84 P.AHY: .ASCII <9>\mbx - device is a mailbox\<0><0>
        00 00 78 6F 62 6C 69 61 6D 20 61 20 73 00E93
        010E001A 00EA0 P.AHX: .LONG 17694746
        00000000 00EA4 .ADDRESS P.AHY
6D 20 65 63 69 76 65 64 20 2D 20 74 6D 64 09 00EAB P.AIA: .ASCII <9>\dmt - device marked for dismount\<0>
6F 6D 73 69 64 20 72 6F 66 20 64 65 6B 72 61 00EB7
        00 74 6E 75 00EC6
        00 00 00ECA .ASCII <0><0>
        010E0021 00ECC P.AHZ: .LONG 17694753
        00000000 00ED0 .ADDRESS P.AIA
68 20 65 63 69 76 65 64 20 2D 20 67 6C 65 09 00ED4 P.AIC: .ASCII <9>\elg - device has error logging ena\
6E 69 67 67 6F 6C 20 72 6F 72 72 65 20 73 61 00EE3
        61 6E 65 20 67 00EF2
        00 64 65 6C 62 00EF7
        010E0027 00EFC P.AIB: .ASCII \bled\<0>
        00000000 00F00 .LONG 17694759
        00F04 P.AIE: .ADDRESS P.AIC
69 20 65 63 69 76 65 64 20 2D 20 6C 6C 61 09 00F04 P.AIE: .ASCII <9>\all - device is allocated\<0><0>
        00 00 64 65 74 61 63 6F 6C 6C 61 20 73 00F13
        010E001A 00F20 P.AID: .LONG 17694746
        00000000 00F24 .ADDRESS P.AIE
69 20 65 63 69 76 65 64 20 2D 20 72 6F 66 09 00F28 P.AIG: .ASCII <9>\for - device is mounted foreign (i\
69 65 72 6F 66 20 64 65 74 6F 75 6F 6D 20 73 00F37
        69 28 20 6E 67 00F46
73 20 65 6C 69 66 2D 6E 6F 6E 20 2C 2E 65 2E 00F4B .ASCII \.e., non-file structured)\
        29 64 65 72 75 74 63 75 72 74 00F5A
        010E003C 00F64 P.AIF: .LONG 17694780
        00000000 00F68 .ADDRESS P.AIG
69 20 65 63 69 76 65 64 20 2D 20 6C 77 73 09 00F6C P.AII: .ASCII <9>\swl - device is software write loc\
74 69 72 77 20 65 72 61 77 74 66 6F 73 20 73 00F7B
        63 6F 6C 20 65 00F8A
        00 00 64 65 6B 00F8F
        010E0026 00F94 P.AIH: .ASCII \ked\<0><0>
        00000000 00F98 .LONG 17694758
        00F9C P.AIK: .ADDRESS P.AII
63 20 65 63 69 76 65 64 20 2D 20 76 64 69 09 00F9C P.AIK: .ASCII <9>\idv - device capable of providing \
69 76 6F 72 70 20 66 6F 20 65 6C 62 61 70 61 00FAB
        20 67 6E 69 64 00FBA
        74 75 70 6E 69 00FBF
        010E0028 00FC4 P.AIJ: .ASCII \input\
        00000000 00FC8 .LONG 17694760
        00FCC P.AIM: .ADDRESS P.AIK
63 20 65 63 69 76 65 64 20 2D 20 76 64 6F 09 00FCC P.AIM: .ASCII <9>\odv - device capable of providing \
69 76 6F 72 70 20 66 6F 20 65 6C 62 61 70 61 00FDB
        20 67 6E 69 64 00FEA
        00 00 00 74 75 70 74 75 6F 00FEF
        010E0029 00FF8 P.AIL: .ASCII \output\<0><0><0>
        00000000 00FFC .LONG 17694761
        00FFC P.AIO: .ADDRESS P.AIM
61 20 65 63 69 76 65 64 20 2D 20 64 6E 72 09 01000 P.AIO: .ASCII <9>\rnd - device allows random access\<0>
63 61 20 6D 6F 64 6E 61 72 20 20 64 6E 73 77 6F 6C 6C 0100F
        00 73 73 65 63 0101E
        00 01023 .ASCII <0>

```



07	05	62 AC	0000'	01 06	FB E1	0007C 0007F	10\$:	CALLS BBC	#1, LIB\$PUT_OUTPUT #6, DEV+1, T1\$	0615 0617
		62		01	FB	00088		PUSHAB	P.AHL	
			05	01	FB	00088	11\$:	CALLS	#1, LIB\$PUT_OUTPUT	0618
				07	18	0008E		TSTB	DEV+1	
			0000'	07	18	0008E		BGEQ	12\$	0620
		62		01	FB	00094		PUSHAB	P.AHN	
		07		01	FB	00094	12\$:	CALLS	#1, LIB\$PUT_OUTPUT	0621
			0000'	06	AC	E9 00097		BLBC	DEV+2, 13\$	0623
				01	FB	0009F		PUSHAB	P.AHP	
07	06	62 AC	0000'	01	FB	0009F	13\$:	CALLS	#1, LIB\$PUT_OUTPUT	0624
				01	E1	000A2		BBC	#1, DEV+2, T4\$	0626
				01	FB	000AB		PUSHAB	P.AHR	
07	06	62 AC	0000'	02	E1	000AE	14\$:	CALLS	#1, LIB\$PUT_OUTPUT	0627
				01	FB	000B3		BBC	#2, DEV+2, T5\$	0629
				01	FB	000B7		PUSHAB	P.AHT	
07	06	62 AC	0000'	03	E1	000BA	15\$:	CALLS	#1, LIB\$PUT_OUTPUT	0630
				01	FB	000C3		BBC	#3, DEV+2, T6\$	0632
				01	FB	000C3		PUSHAB	P.AHV	
07	06	62 AC	0000'	04	E1	000C6	16\$:	CALLS	#1, LIB\$PUT_OUTPUT	0633
				01	FB	000C6		BBC	#4, DEV+2, T7\$	0635
				01	FB	000CB		PUSHAB	P.AHX	
07	06	62 AC	0000'	01	FB	000CF	17\$:	CALLS	#1, LIB\$PUT_OUTPUT	0636
				05	E1	000D2		BBC	#5, DEV+2, T8\$	0638
				01	FB	000DB		PUSHAB	P.AHZ	
07	06	62 AC	0000'	01	FB	000DB	18\$:	CALLS	#1, LIB\$PUT_OUTPUT	0639
				06	E1	000DE		BBC	#6, DEV+2, T9\$	0641
				01	FB	000E3		PUSHAB	P.AIB	
		62		01	FB	000E7		CALLS	#1, LIB\$PUT_OUTPUT	
			06	06	AC	95 000EA	19\$:	TSTB	DEV+2	0642
				07	18	000ED		BGEQ	20\$	
			0000'	01	FB	000F3		PUSHAB	P.AID	0644
		62		01	FB	000F3		CALLS	#1, LIB\$PUT_OUTPUT	
		07		07	AC	E9 000F6	20\$:	BLBC	DEV+3, 21\$	0645
				01	FB	000FA		PUSHAB	P.AIF	0647
				01	FB	000FE		CALLS	#1, LIB\$PUT_OUTPUT	
07	07	62 AC	0000'	01	E1	00101	21\$:	BBC	#1, DEV+3, 22\$	0648
				01	FB	0010A		PUSHAB	P.AIH	0650
				02	E1	0010D	22\$:	CALLS	#1, LIB\$PUT_OUTPUT	
07	07	62 AC	0000'	02	E1	0010D	22\$:	BBC	#2, DEV+3, 23\$	0651
				01	FB	00116		PUSHAB	P.AIJ	0653
				03	E1	00119	23\$:	CALLS	#1, LIB\$PUT_OUTPUT	
07	07	62 AC	0000'	03	E1	00119	23\$:	BBC	#3, DEV+3, 24\$	0654
				01	FB	00122		PUSHAB	P.AIL	0656
				04	E1	00125	24\$:	CALLS	#1, LIB\$PUT_OUTPUT	
07	07	62 AC	0000'	04	E1	00125	24\$:	BBC	#4, DEV+3, 25\$	0657
				01	FB	0012E		PUSHAB	P.AIN	0659
				05	E1	00131	25\$:	CALLS	#1, LIB\$PUT_OUTPUT	
07	07	62 AC	0000'	05	E1	00131	25\$:	BBC	#5, DEV+3, 26\$	0660
				01	FB	0013A		PUSHAB	P.AIP	0662
				06	E1	0013D	26\$:	CALLS	#1, LIB\$PUT_OUTPUT	
07	07	62 AC	0000'	06	E1	0013D	26\$:	BBC	#6, DEV+3, 27\$	0663
				01	FB	00142		PUSHAB	P.AIR	0665
				01	FB	00146		CALLS	#1, LIB\$PUT_OUTPUT	
		62		07	AC	95 00149	27\$:	TSTB	DEV+3	0666
				07	18	0014C		BGEQ	28\$	
			0000'	01	FB	0014E		PUSHAB	P.AIT	0668







The image displays a grid of 100 small terminal window screenshots, arranged in a 10x10 pattern. Each window shows a different VAX/VMS command and its output. The windows are arranged in a 10x10 grid. Some windows have larger text overlays, such as 'EXCDEF5 LIS', 'EXCDIRE LIS', 'EXCDBG LIS', and 'EXCDS11 LIS'. The screenshots show various system commands, file listings, and error messages.