



DDDDDDDDDDDD		UUU	UUU	MMM	MMM	PPPPPPPPPP	
DDDDDDDDDDDD		UUU	UUU	MMM	MMM	PPPPPPPPPP	
DDDDDDDDDDDD		UUU	UUU	MMM	MMM	PPPPPPPPPP	
DDD	DDD	UUU	UUU	MMMMMM	MMMMMM	PPP	PPP
DDD	DDD	UUU	UUU	MMMMMM	MMMMMM	PPP	PPP
DDD	DDD	UUU	UUU	MMMMMM	MMMMMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDD	DDD	UUU	UUU	MMM	MMM	PPP	PPP
DDDDDDDDDDDD		UUUUUUUUUUUU	UUUUUUUUUUUU	MMM	MMM	PPP	
DDDDDDDDDDDD		UUUUUUUUUUUU	UUUUUUUUUUUU	MMM	MMM	PPP	
DDDDDDDDDDDD		UUUUUUUUUUUU	UUUUUUUUUUUU	MMM	MMM	PPP	

```

DDDDDDDD UU  UU MM  MM PPPPPPP HH  HH EEEEEEEEE AAAAAA DDDDDDD EEEEEEEEE
DDDDDDDD UU  UU MM  MM PPPPPPP HH  HH EEEEEEEEE AAAAAA DDDDDDD EEEEEEEEE
DD  DD UU  UU MMMM MMMM PP  PP HH  HH EE  AA  AA DD  DD EE
DD  DD UU  UU MMMM MMMM PP  PP HH  HH EE  AA  AA DD  DD EE
DD  DD UU  UU MM  MM PP  PP HH  HH EE  AA  AA DD  DD EE
DD  DD UU  UU MM  MM PPPPPPP HHHHHHHHHH EEEEEEE AA  AA DD  DD EEEEEEE
DD  DD UU  UU MM  MM PPPPPPP HHHHHHHHHH EEEEEEE AA  AA DD  DD EEEEEEE
DD  DD UU  UU MM  MM PP  HH  HH EE  AAAAAAAAAA DD  DD EE
DD  DD UU  UU MM  MM PP  HH  HH EE  AAAAAAAAAA DD  DD EE
DD  DD UU  UU MM  MM PP  HH  HH EE  AA  AA DD  DD EE
DD  DD UU  UU MM  MM PP  HH  HH EE  AA  AA DD  DD EE
DD  DD UU  UU MM  MM PP  HH  HH EEEEEEEEE AA  AA DDDDDDD EEEEEEEEE
DDDDDDDD UUUUUUUUUU MM  MM PP  HH  HH EEEEEEEEE AA  AA DDDDDDD EEEEEEEEE
DDDDDDDD UUUUUUUUUU MM  MM PP  HH  HH EEEEEEEEE AA  AA DDDDDDD EEEEEEEEE

```

```

LL  IIIII SSSSSSS
LL  IIIII SSSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SSSSSS
LL  II  SSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLL IIIII SSSSSSS
LLLLLLLLLL IIIII SSSSSSS

```

D V

.....

```

1 0001 0 MODULE DUMP$HEADER (
2 0002 0     IDENT='V04-000',
3 0003 0     ADDRESSING_MODE(EXTERNAL=GENERAL,
4 0004 0     NONEXTERNAL=LONG_RELATIVE)
5 0005 0     ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1
9 0009 1 *****
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1 * ALL RIGHTS RESERVED.
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
20 0020 1 * TRANSFERRED.
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
24 0024 1 * CORPORATION.
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
28 0028 1 *
29 0029 1 *
30 0030 1 *****
31 0031 1
32 0032 1
33 0033 1 ++
34 0034 1
35 0035 1 FACILITY: File dump utility
36 0036 1
37 0037 1 ABSTRACT:
38 0038 1     This module contains the routines that dump file headers.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1     VAX native, user mode.
42 0042 1
43 0043 1 --
44 0044 1
45 0045 1
46 0046 1 AUTHOR: Martin L. Jack          CREATION DATE: 16-Aug-1981
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1     V03-006 LMP0176          L. Mark Pilant,          6-Dec-1983  9:52
51 0051 1     Use the correct display width in the call to $FORMAT_ACL.
52 0052 1     Also, fix a bug in the code that displays the file name
53 0053 1     extension.
54 0054 1
55 0055 1     V03-005 LMP0128          L. Mark Pilant,          13-Jul-1983 11:56
56 0056 1     Track changes made to $FH2DEF and friends.
57 0057 1

```

.. 58 0058 1 !
.. 59 0059 1 !
.. 60 0060 1 !
.. 61 0061 1 !
.. 62 0062 1 !
.. 63 0063 1 !
.. 64 0064 1 !
.. 65 0065 1 !
.. 66 0066 1 !
.. 67 0067 1 !
.. 68 0068 1 !
.. 69 0069 1 !
.. 70 0070 1 !
.. 71 0071 1 !
.. 72 0072 1 !
.. 73 0073 1 !
.. 74 0074 1 !
.. 75 0075 1 !
.. 76 0076 1 !
.. 77 0077 1 !
.. 78 0078 1 !
.. 79 0079 1 !
.. 80 0080 1 !--

V03-004 LMP0100 L. Mark Pilant, 14-Apr-1983 8:47
Convert to the new system service \$FORMAT_ACL.
V03-003 ACG0325 Andrew C. Goldstein, 4-Apr-1983 17:38
Fix file header area length check
V03-002 LMP0050 L. Mark Pilant, 8-Oct-1982 12:50
Add the ability to dump the ACL if it exists.
V03-001 MLJ0090 Martin L. Jack, 6-May-1982 7:52
Avoid clobbering checksum word for blocks that are not file
headers.
V03-003 MLJ0080 Martin L. Jack, 17-Feb-1982 16:04
Add FAT\$W_GBC.
V03-002 MLJ0045 Martin L. Jack, 12-Sep-1981 17:00
Allow file header dump for tape.
V03-001 MLJ0041 Martin L. Jack, 3-Sep-1981 21:33
Add FCH\$V_ERASE.

```

82      0081 1 LIBRARY 'SYSSLIBRARY:LIB.L32';
83      0082 1 REQUIRE 'SRCS:DUMPRE';
84      0198 1
85      0199 1
86      0200 1 FORWARD ROUTINE
87      0201 1 DUMPSHEADER: NOVALUE,      ! Dump header(s) for current input file
88      0202 1 CHECK_HEADER,      ! Validate file header
89      0203 1 DUMPSFAO: NOVALUE,      ! Format information into line
90      0204 1 DUMPSEOL: NOVALUE,      ! Write line buffer
91      0205 1 DUMSPUT_FAO: NOVALUE,      ! Format information and write buffer
92      0206 1 FORMAT_BIT_STRING:      !
93      0207 1 NOVALUE,      ! Format a bit string
94      0208 1 FORMAT_CODED_VALUE:      !
95      0209 1 NOVALUE,      ! Format a coded value
96      0210 1 FORMAT_DATE_VALUE:      !
97      0211 1 NOVALUE,      ! Format a standard system date
98      0212 1 FORMAT_ODS1_DATE_VALUE:      !
99      0213 1 NOVALUE,      ! Format an ODS-1 date
100     0214 1 FORMAT_PROTECTION:      !
101     0215 1 NOVALUE,      ! Format a protection value
102     0216 1 FORMAT_RMS_ATTRIBUTES:      !
103     0217 1 NOVALUE,      ! Format RMS attributes area
104     0218 1 DUMPSONE_HEADER;      ! Dump block as a header, if valid
105     0219 1
106     0220 1
107     0221 1 EXTERNAL ROUTINE
108     0222 1 CHECKSUM,      ! Compute file header checksum
109     0223 1 MAKE_STRING,      ! Convert ODS-1 filename to ASCII
110     0224 1 DUMPSBLANK_LINE:NOVALUE,      ! Write blank line
111     0225 1 DUMPSDUMP_BUFFER:NOVALUE,      ! Dump one buffer
112     0226 1 DUMPSNEW_PAGE:NOVALUE,      ! Force page break
113     0227 1 DUMPSOUTPUT_GETMSG:NOVALUE,      ! Write gotten message
114     0228 1 DUMSPUT_LINE:NOVALUE;      ! Write line
115     0229 1
116     0230 1
117     0231 1 EXTERNAL
118     0232 1 DUMPSGL_FLAGS: BBLOCK,      ! General flags
119     0233 1 DUMPSGL_IDESC: BBLOCK,      ! Descriptor for input filename
120     0234 1 DUMPSGL_INAM: REF BBLOCK,      ! Input NAM block
121     0235 1 DUMPSGL_OUTDESC:BBLOCK,      ! Descriptor for output buffer
122     0236 1 DUMPSGL_WIDTH;      ! Width of output medium
123     0237 1
124     0238 1
125     0239 1 EXTERNAL LITERAL
126     0240 1 DUMPS_HEADER,
127     0241 1 DUMPS_READHEADER;
128     0242 1
129     0243 1
130     0244 1 OWN
131     0245 1 LINE_DESC: VECTOR[2];
132     0246 1
133     0247 1
134     0248 1 MACRO
135     M 0249 1 DUMSPUT_FAO(A)=
136     M 0250 1 DUMSPUT_FAO (UPLIT BYTE (%ASCIC A)
137     0251 1 %IF NOT %NULL(%REMAINING) %THEN , %FI %REMAINING) %,
138     0252 1

```

DUMPSHEADER
V04-000

D 16
16-Sep-1984 01:30:28
14-Sep-1984 12:21:37

VAX-11 Bliss-32 V4.0-742
[DUMP.SRC]DUMPHEADE.B32;1

Page 4
(2)

: 139 M 0253 1
: 140 M 0254 1
: 141 0255 1

DUMPSFAO(A)=
DUMPSFAO (UPLIT BYTE (%ASCIC A)
%IF NOT %NULL(%REMAINING) %THEN , %FI %REMAINING) %.

```

: 143      0256 1 GLOBAL ROUTINE DUMPSHEADER: NOVALUE=
: 144      0257 1
: 145      0258 1 |++
: 146      0259 1
: 147      0260 1 |
: 148      0261 1 | FUNCTIONAL DESCRIPTION:
: 149      0262 1 |   This routine dumps the file header(s) for the current input file.
: 150      0263 1 |
: 151      0264 1 | INPUT PARAMETERS:
: 152      0265 1 |   NONE
: 153      0266 1 |
: 154      0267 1 | IMPLICIT INPUTS:
: 155      0268 1 |   NONE
: 156      0269 1 |
: 157      0270 1 | OUTPUT PARAMETERS:
: 158      0271 1 |   NONE
: 159      0272 1 |
: 160      0273 1 | IMPLICIT OUTPUTS:
: 161      0274 1 |   NONE
: 162      0275 1 |
: 163      0276 1 | ROUTINE VALUE:
: 164      0277 1 |   NONE
: 165      0278 1 |
: 166      0279 1 | SIDE EFFECTS:
: 167      0280 1 |   NONE
: 168      0281 1 | --
: 169      0282 1
: 170      0283 2 BEGIN
: 171      0284 2 LOCAL
: 172      0285 2     NAM:          REF BBLOCK,      ! Pointer to input file NAM block
: 173      0286 2     DESC:          VECTOR[2],      ! Utility descriptor
: 174      0287 2     HEADER_DESC:  VECTOR[2],      ! Header block descriptor
: 175      0288 2     CHANNEL:      WORD,          ! Channel assigned to input device
: 176      0289 2     FIB:          BBLOCK[10],     ! FIB
: 177      0290 2     ATR_DESC:     BBLOCK[12],     ! ACP attributes list
: 178      0291 2     HEADER:      BBLOCK[512],    ! File header
: 179      0292 2     VOLATILE,
: 180      0293 2     STATUS,      ! Status variable
: 181      0294 2     IOSB:      VECTOR[4,WORD]; ! I/O status block
: 182      0295 2
: 183      0296 2
: 184      0297 2 ! Get a descriptor for the device name and assign a channel to the device.
: 185      0298 2 !
: 186      0299 2 NAM = DUMPSGL_INAM;
: 187      0300 2 DESC[0] = .NAM[NAM$B_DEV];
: 188      0301 2 DESC[1] = .NAM[NAM$L_DEV];
: 189      0302 2 STATUS = $ASSIGN(DEVNAM=DESC, CHAN=CHANNEL);
: 190      0303 2 IF NOT .STATUS
: 191      0304 2 THEN
: 192      0305 2     BEGIN
: 193      0306 2     SIGNAL(DUMPS_READHEADER, 1, DUMPSGL_IDESC, .STATUS);
: 194      0307 2     RETURN;
: 195      0308 2     END;
: 196      0309 2
: 197      0310 2
: 198      0311 2 ! Initialize to read the primary file header.
: 199      0312 2 !
```

```

200 0313 2 DESC[0] = 10;
201 0314 2 DESC[1] = FIB;
202 0315 2 FIB[FIB$L_ACCTL] = 0;
203 0316 2 FIB[FIB$W_FID_NUM] = .NAM[NAM$W_FID_NUM];
204 0317 2 FIB[FIB$W_FID_SEQ] = .NAM[NAM$W_FID_SEQ];
205 0318 2 FIB[FIB$W_FID_RVN] = .NAM[NAM$W_FID_RVN];
206 0319 2 ATR_DESC[ATR$Q_SIZE] = ATR$S_HEADER;
207 0320 2 ATR_DESC[ATR$W_TYPE] = ATR$C_HEADER;
208 0321 2 ATR_DESC[ATR$L_ADDR] = HEADER;
209 0322 2 ATR_DESC[8,0,32,0] = 0;
210 0323 2
211 0324 2
212 0325 2 ! Loop until there are no more headers to be dumped.
213 0326 2 !
214 0327 2 WHILE TRUE DO
215 0328 2 BEGIN
216 0329 2
217 0330 2 ! Read the file header.
218 0331 2 !
219 0332 2 STATUS = $QIOW(
220 0333 2 FUNC=IOS_ACCESS,
221 0334 2 CHAN=.CHANNEL,
222 0335 2 IOSB=IOSB,
223 0336 2 P1=DESC,
224 0337 2 P5=ATR_DESC);
225 0338 2 IF .STATUS THEN STATUS = .IOSB[0];
226 0339 2 IF NOT .STATUS
227 0340 2 THEN
228 0341 2 BEGIN
229 0342 2 SIGNAL(DUMPS_READHEADER, 1, DUMPS$GL_IDESC, .STATUS);
230 0343 2 EXITLOOP;
231 0344 2 END;
232 0345 2
233 0346 2
234 0347 2 ! Dump the file header that was just read.
235 0348 2 !
236 0349 2 IF .DUMPS$GL_FLAGS[DUMPSV_FORMATTED]
237 0350 2 THEN
238 0351 2 BEGIN
239 0352 2 DUMPS$NEW_PAGE();
240 0353 2 DUMPS$OUTPUT_GETMSG(DUMPS_HEADER, %B'0001');
241 0354 2 DUMPS$BLANK [LINE()];
242 0355 2 DUMPS$ONE_HEADER(HEADER);
243 0356 2 END
244 0357 2 ELSE
245 0358 2 BEGIN
246 0359 2 HEADER_DESC[0] = 512;
247 0360 2 HEADER_DESC[1] = HEADER;
248 0361 2 DUMPS$DUMP_BUFFER(HEADER_DESC, SSS_NORMAL, TRUE);
249 0362 2 END;
250 0363 2
251 0364 2
252 0365 2 ! Get the file ID of the next extension header, if any.
253 0366 2 !
254 0367 2 IF .HEADER[FH2$B_STRUCLEV] EQL 2
255 0368 2 THEN
256 0369 2 BEGIN

```



```

: 257 0370 4 FIB[FIBSW_FID_NUM] = .HEADER[FH2$W_EX_FIDNUM];
: 258 0371 4 FIB[FIBSW_FID_SEQ] = .HEADER[FH2$W_EX_FIDSEQ];
: 259 0372 4 FIB[FIBSB_FID_NMX] = .HEADER[FH2$B_EX_FIDNMX];
: 260 0373 4 IF .HEADER[FH2$B_EX_FIDRVN] NEQ 0
: 261 0374 4 THEN FIB[FIBSB_FID_RVN] = .HEADER[FH2$B_EX_FIDRVN];
: 262 0375 4 END
: 263 0376 3 ELSE
: 264 0377 4 BEGIN LOCAL MAP AREA: REF BBLOCK;
: 265 0378 4 MAP_AREA = HEADER + .HEADER[FH1$B_MPOFFSET]*2;
: 266 0379 4 FIB[FIBSW_FID_NUM] = .MAP_AREA[FM1$W_EX_FILNUM];
: 267 0380 4 FIB[FIBSW_FID_SEQ] = .MAP_AREA[FM1$W_EX_FILSEQ];
: 268 0381 3 END;
: 269 0382 3
: 270 0383 3
: 271 0384 3 ! If no more extension headers, we are done.
: 272 0385 3
: 273 0386 3 IF .FIB[FIBSW_FID_NUM] EQL 0 THEN EXITLOOP;
: 274 0387 2 END;
: 275 0388 2
: 276 0389 2
: 277 0390 2 ! All done. Deassign the channel and return.
: 278 0391 2
: 279 0392 2 $DASSGN(CHAN=.CHANNEL);
: 280 0393 1 END;

```

```

.TITLE DUMPS$HEADER
.IDENT \V04-000\

.PSECT $OWNS,NOEXE,2

0000 LINE_DESC:
.BLKB 8

.EXTRN CHECKSUM, MAKE STRING
.EXTRN DUMPS$BLANK_LINE
.EXTRN DUMPS$DUMP_BUFFER
.EXTRN DUMPS$NEW_PAGE, DUMPS$OUTPUT_GETMSG
.EXTRN DUMPS$PUT_LINE, DUMPS$GL_FLAGS
.EXTRN DUMPS$GL_IDESC, DUMPS$GL_INAM
.EXTRN DUMPS$GL_OUTDESC
.EXTRN DUMPS$GL_WIDTH, DUMPS$HEADER
.EXTRN DUMPS$READHEADER
.EXTRN SYSS$ASSIGN, SYSS$QIOW
.EXTRN SYSS$DASSGN

.PSECT $CODES,NOWRT,2

.ENTRY DUMPS$HEADER, Save R2,R3,R4,R5,R6 ; 0256
MOVAB LIB$SIGNAL, R6
MOVL #DUMPS$READHEADER, R5
MOVAB DUMPS$GL_IDESC, R4
MOVAB -564(SP), SP
MOVL DUMPS$GL_INAM, NAM ; 0299
MOVZBL 57(NAM), DESC ; 0300
MOVL 68(NAM), DESC+4 ; 0301
CLRQ -(SP) ; 0302

```

```

007C 00000
56 00000000G 00 9E 00002
55 00000000G 8F D0 00009
54 00000000G 00 9E 00010
5E FDCC CE 9E 00017
52 00000000G 00 D0 0001C
F8 AD 39 A2 9A 00023
FC AD 44 A2 D0 00028
7E 7C 0002D

```

			08	AE	9F	0002F		PUSHAB	CHANNEL		
			F8	AD	9F	00032		PUSHAB	DESC		
00000000G	00		04	FB	00035			CALLS	#4, SYSS\$ASSIGN		
	53		50	DD	0003C			MOVL	R0, STATUS		
	0C		53	EB	0003F			BLBS	STATUS, 1\$		0303
			53	DD	00042			PUSHL	STATUS		0306
			54	DD	00044			PUSHL	R4		
			01	DD	00046			PUSHL	#1		
			55	DD	00048			PUSHL	R5		
	66		04	FB	0004A			CALLS	#4, LIB\$\$SIGNAL		
			04	00	0004D			RET			0305
F8	AD		0A	DD	0004E	1\$:		MOVL	#10, DESC		0313
FC	AD		E4	AD	9E	00052		MOVAB	FIB, DESC+4		0314
EB	AD		24	A2	DD	00057		MOVL	36(NAM), FIB+4		0316
EC	AD		28	A2	BD	0005C		MOVW	40(NAM), FIB+8		0318
DB	AD	000A0200	8F	DD	00061			MOVL	#655872, ATR_DESC		0319
DC	AD		0C	AE	9E	00069		MOVAB	HEADER, ATR_DESC+4		0321
			E0	AD	7C	0006E		CLRQ	ATR_DESC+8		0322
			7E	D4	00071	2\$:		CLRL	-(SP)		0337
			D8	AD	9F	00073		PUSHAB	ATR_DESC		
			7E	7C	00076			CLRQ	-(SP)		
			7E	D4	00078			CLRL	-(SP)		
			F8	AD	9F	0007A		PUSHAB	DESC		
			7E	7C	0007D			CLRQ	-(SP)		
			24	AE	9F	0007F		PUSHAB	IOSB		
			32	DD	00082			PUSHL	#50		
	7E		28	AE	3C	00084		MOVZWL	CHANNEL, -(SP)		
			7E	D4	00088			CLRL	-(SP)		
00000000G	00		0C	FB	0008A			CALLS	#12, SYSS\$QIOW		
	53		50	DD	00091			MOVL	R0, STATUS		
	07		53	E9	00094			BLBC	STATUS, 3\$		0338
	53		04	AE	3C	00097		MOVZWL	IOSB, STATUS		
	0E		53	EB	0009B			BLBS	STATUS, 4\$		0339
			53	DD	0009E	3\$:		PUSHL	STATUS		0342
			54	DD	000A0			PUSHL	R4		
			01	DD	000A2			PUSHL	#1		
			55	DD	000A4			PUSHL	R5		
	66		04	FB	000A6			CALLS	#4, LIB\$\$SIGNAL		
			0081	31	000A9			BRW	9\$		0341
29 00000000G	00		05	E1	000AC	4\$:		BBC	#5, DUMPS\$GL_FLAGS, 5\$		0349
00000000G	00		00	FB	000B4			CALLS	#0, DUMPS\$NEW_PAGE		0352
			01	DD	000BB			PUSHL	#1		0353
		00000000G	8F	DD	000BD			PUSHL	#DUMPS\$HEADER		
00000000G	00		02	FB	000C3			CALLS	#2, DUMPS\$OUTPUT_GETMSG		
00000000G	00		00	FB	000CA			CALLS	#0, DUMPS\$BLANK_LINE		0354
			0C	AE	9F	000D1		PUSHAB	HEADER		0355
00000000V	EF		01	FB	000D4			CALLS	#1, DUMPS\$ONE_HEADER		
			19	11	000DB			BRB	6\$		0349
	F0	AD	0200	8F	3C	000DD	5\$:	MOVZWL	#512, HEADER_DESC		0359
	F4	AD	0C	AE	9E	000E3		MOVAB	HEADER, HEADER_DESC+4		0360
			01	DD	000E8			PUSHL	#1		0361
			01	DD	000EA			PUSHL	#1		
00000000G	00		F0	AD	9F	000EC		PUSHAB	HEADER_DESC		
	02		03	FB	000EF			CALLS	#3, DUMPS\$DUMP_BUFFER		
			13	AE	91	000F6	6\$:	CMPB	HEADER+7, #2		0367
			1B	12	000FA			BNEQ	7\$		
	E8	AD	1A	AE	B0	000FC		MOVW	HEADER+14, FIB+4		0370

DUMPSHEADER
V04-000

I 16
16-Sep-1984 01:30:28 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:21:37 [DUMP.SRC]DUMPHEADE.B32;1

Page 9
(3)

EA	AD	1C	AE	B0	00101	MOVW	HEADER+16, FIB+6	:	0371
ED	AD	1F	AE	90	00106	MOVW	HEADER+19, FIB+9	:	0372
		1E	AE	95	0010B	TSTB	HEADER+18	:	0373
			15	13	0010E	BEQL	8\$:	
EC	AD	1E	AE	90	00110	MOVW	HEADER+18, FIB+8	:	0374
			0E	11	00115	BRB	8\$:	0367
	50	0D	AE	9A	00117	MOVZBL	HEADER+1, R0	:	0378
	50	0C	AE40	3E	0011B	MOVAW	HEADER[R0], MAP AREA	:	
E8	AD	02	A0	D0	00120	MOVL	2(MAP_AREA), FIB+4	:	0379
		E8	AD	B5	00125	TSTW	FIB+4	:	0386
			03	13	00128	BEQL	9\$:	
			FF44	31	0012A	BRW	2\$:	
	7E		6E	3C	0012D	MOVZWL	CHANNEL, -(SP)	:	0392
00000000G	00		01	FB	00130	CALLS	#1, SYSSDASSGN	:	
			04	00	00137	RET		:	0393

; Routine Size: 312 bytes, Routine Base: \$CODE\$ + 0000

```

: 282      0394 1 ROUTINE CHECK_HEADER(HEADER)=
: 283      0395 1
: 284      0396 1 !++
: 285      0397 1
: 286      0398 1 FUNCTIONAL DESCRIPTION:
: 287      0399 1     This routine validates a block as a file header.
: 288      0400 1
: 289      0401 1 INPUT PARAMETERS:
: 290      0402 1     HEADER           - Pointer to block.
: 291      0403 1
: 292      0404 1 IMPLICIT INPUTS:
: 293      0405 1     NONE
: 294      0406 1
: 295      0407 1 OUTPUT PARAMETERS:
: 296      0408 1     NONE
: 297      0409 1
: 298      0410 1 IMPLICIT OUTPUTS:
: 299      0411 1     NONE
: 300      0412 1
: 301      0413 1 ROUTINE VALUE:
: 302      0414 1     0 if the block is not a valid header
: 303      0415 1     1 if the block is a valid ODS-1 header
: 304      0416 1     2 if the block is a valid ODS-2 header
: 305      0417 1
: 306      0418 1 SIDE EFFECTS:
: 307      0419 1     NONE
: 308      0420 1
: 309      0421 1 --
: 310      0422 1
: 311      0423 2 BEGIN
: 312      0424 2 MAP
: 313      0425 2     HEADER:           REF BBLOCK:           ! Pointer to file header
: 314      0426 2 LOCAL
: 315      0427 2     SAVE_CHECK;           ! Save checksum word
: 316      0428 2
: 317      0429 2
: 318      0430 2 ! Ensure that the checksum is correct.
: 319      0431 2
: 320      0432 2 SAVE_CHECK = .HEADER[FH2$W_CHECKSUM];
: 321      0433 2 IF NOT CHECKSUM(.HEADER)
: 322      0434 2 THEN
: 323      0435 3     BEGIN
: 324      0436 3     HEADER[FH2$W_CHECKSUM] = .SAVE_CHECK;
: 325      0437 3     RETURN 0;
: 326      0438 2     END;
: 327      0439 2
: 328      0440 2
: 329      0441 2 ! Basic validity test for an ODS-2 header.
: 330      0442 2
: 331      0443 2 IF .HEADER[FH2$B_STRUCLEV] EQL 2
: 332      0444 2 THEN
: 333      0445 3     BEGIN
: 334      0446 3     IF
: 335      0447 3     .HEADER[FH2$B_IDOFFSET] LSSU $BYTEOFFSET (FH2$L HIGHWATER)/2 OR
: 336      0448 3     .HEADER[FH2$B_MPOFFSET] LSSU .HEADER[FH2$B_IDOFFSET] OR
: 337      0449 3     .HEADER[FH2$B_ACOFFSET] LSSU .HEADER[FH2$B_MPOFFSET] OR
: 338      0450 3     .HEADER[FH2$B_RSOFFSET] LSSU .HEADER[FH2$B_ACOFFSET] OR

```

```

: 339 0451 3 .HEADER[FH2$B_MAP_INUSE] GTRU .HEADER[FH2$B_ACOFFSET] - .HEADER[FH2$B_MPOFFSET]
: 340 0452 THEN
: 341 0453 RETURN 0
: 342 0454 ELSE
: 343 0455 RETURN 2;
: 344 0456 END;
: 345 0457
: 346 0458
: 347 0459 ! Basic validity test for an ODS-1 header.
: 348 0460
: 349 0461 IF .HEADER[FH2$B_STRUCLEV] EQL 1
: 350 0462 THEN
: 351 0463 BEGIN
: 352 0464 LOCAL
: 353 0465 MAP_AREA: REF BBLOCK; ! Pointer to map area
: 354 0466
: 355 0467
: 356 0468 ! Check the area offsets, the extension RVN, and the retrieval pointer
: 357 0469 ! data for consistency.
: 358 0470
: 359 0471 IF .HEADER[FH1$B_IDOFFSET] NEQ FH1$C_LENGTH / 2
: 360 0472 THEN
: 361 0473 RETURN 0;
: 362 0474
: 363 0475
: 364 0476 MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
: 365 0477 IF
: 366 0478 .MAP_AREA[FM1$B_EX RVN] NEQ 0 OR
: 367 0479 .MAP_AREA[FM1$B_COUNTSIZE] NEQ 1 OR
: 368 0480 .MAP_AREA[FM1$B_LBNSIZE] NEQ 3 OR
: 369 0481 .MAP_AREA[FM1$B_INUSE] GTRU .MAP_AREA[FM1$B_AVAIL] OR
: 370 0482 .MAP_AREA[FM1$B_AVAIL] GTRU 255 = (.MAP_AREA + FM1$C_POINTERS - .HEADER) / 2
: 371 0483 THEN
: 372 0484 RETURN 0
: 373 0485 ELSE
: 374 0486 RETURN 1;
: 375 0487 END;
: 376 0488
: 377 0489
: 378 0490 RETURN 0;
: 379 0491 END;

```

```

000C 0000 CHECK_HEADER:
          52      04 AC D0 00002      .WORD      Save R2,R3      : 0394
          53      01FE C2 3C 00006     MOVL      HEADER, R2      : 0432
00000000G 00          52 DD 0000B     MOVZWL   510(R2), SAVE_CHECK : 0433
          07          01 FB 0000D     PUSHL   R2
          01FE C2          50 E8 00014     CALLS   #1, CHECKSUM
          02          53 B0 00017     BLBS    R0, 1$
          07          7B 11 0001C     MOVW   SAVE_CHECK, 510(R2) : 0436
          02          A2 91 0001E 1$ : BRB     3$ : 0437
          07          30 12 00022     CMPB   7(R2), #2 : 0443
          02          30 12 00022     BNEQ   2$

```

DUMPSHEADER
V04-000

L 16
16-Sep-1984 01:30:28 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:21:37 [DUMP.SRC]DUMPHEAD.B32;1

Page 12
(4)

			26		62	91	00024		CMPB	(R2), #38		0447
					70	1F	00027		BLSSU	3\$		
			62	01	A2	91	00029		CMPB	1(R2), (R2)		0448
					6A	1F	0002D		BLSSU	3\$		
		01	A2	02	A2	91	0002F		CMPB	2(R2), 1(R2)		0449
					63	1F	00034		BLSSU	3\$		
		02	A2	03	A2	91	00036		CMPB	3(R2), 2(R2)		0450
					5C	1F	0003B		BLSSU	3\$		
			50	02	A2	9A	0003D		MOVZBL	2(R2), R0		0451
				01	A2	9A	00041		MOVZBL	1(R2), R1		
			50		51	C2	00045		SUBL2	R1, R0		
50		3A	A2		08	00	ED 00048		CMPZV	#0, #8, 58(R2), R0		
					49	1A	0004E		BGTRU	3\$		
			50		02	D0	00050		MOVL	#2, R0		0455
						04	00053		RET			
			01	07	A2	91	00054	2\$:	CMPB	7(R2), #1		0461
					3F	12	00058		BNEQ	3\$		
			17		62	91	0005A		CMPB	(R2), #23		0471
					3A	12	0005D		BNEQ	3\$		
			50	01	A2	9A	0005F		MOVZBL	1(R2), R0		0476
			50		6240	3E	00063		MOVAV	(R2)[R0], MAP_AREA		
				01	A0	95	00067		TSTB	1(MAP_AREA)		0478
					2D	12	0006A		BNEQ	3\$		
			01	06	A0	91	0006C		CMPB	6(MAP_AREA), #1		0479
					27	12	00070		BNEQ	3\$		
			03	07	A0	91	00072		CMPB	7(MAP_AREA), #3		0480
					21	12	00076		BNEQ	3\$		
		09	A0	08	A0	91	00078		CMPB	8(MAP_AREA), 9(MAP_AREA)		0481
					1A	1A	0007D		BGTRU	3\$		
			52		50	C2	0007F		SUBL2	MAP_AREA, R2		0482
			52		0A	C2	00082		SUBL2	#10, R2		
			52		02	C6	00085		DIVL2	#2, R2		
			52	00FF	C2	9E	00088		MOVAB	255(R2), R2		
52		09	A0		08	00	ED 0008D		CMPZV	#0, #8, 9(MAP_AREA), R2		
					04	1A	00093		BGTRU	3\$		
			50		01	D0	00095		MOVL	#1, R0		0486
						04	00098		RET			
					50	D4	00099	3\$:	CLRL	R0		0491
					04	0009B			RET			

; Routine Size: 156 bytes, Routine Base: \$CODE\$ + 0138

```

: 381 0492 1 ROUTINE DUMPSFAO_(CTRL,PARAM): NOVALUE=
: 382 0493 1
: 383 0494 1 ++
: 384 0495 1
: 385 0496 1 FUNCTIONAL DESCRIPTION:
: 386 0497 1 This routine interfaces to FAO to format information into the line.
: 387 0498 1
: 388 0499 1 INPUT PARAMETERS:
: 389 0500 1 CTRL - ASCII control string
: 390 0501 1 PARAM... - Parameters required by the control string (if any)
: 391 0502 1
: 392 0503 1 IMPLICIT INPUTS:
: 393 0504 1 NONE
: 394 0505 1
: 395 0506 1 OUTPUT PARAMETERS:
: 396 0507 1 NONE
: 397 0508 1
: 398 0509 1 IMPLICIT OUTPUTS:
: 399 0510 1 Information formatted into the line buffer.
: 400 0511 1
: 401 0512 1 ROUTINE VALUE:
: 402 0513 1 NONE
: 403 0514 1
: 404 0515 1 SIDE EFFECTS:
: 405 0516 1 NONE
: 406 0517 1
: 407 0518 1 --
: 408 0519 1
: 409 0520 2 BEGIN
: 410 0521 2 MAP
: 411 0522 2 CTRL: REF VECTOR[.BYTE]; ! ASCII control string
: 412 0523 2 LOCAL
: 413 0524 2 OUTLEN: WORD ! $FAO resultant length
: 414 0525 2 DESC: VECTOR[2]; ! Control string descriptor
: 415 0526 2
: 416 0527 2
: 417 0528 2 DESC[0] = .CTRL[0];
: 418 0529 2 DESC[1] = CTRL[1];
: 419 0530 2 $FAOL(CTRSTR=DESC, OUTLEN=OUTLEN, OUTBUF=LINE_DESC, PRMLST=PARAM);
: 420 0531 2 LINE_DESC[0] = .LINE_DESC[0] - .OUTLEN;
: 421 0532 2 LINE_DESC[1] = .LINE_DESC[1] + .OUTLEN;
: 422 0533 1 END;

```

.EXTRN SYS\$FAOL

```

                                0004 00000 DUMPSFAO :
                                .WORD Save R2 : 0492
                                MOVAB LINE_DESC, R2 :
                                SUBL2 #12, SP :
                                MOVZBL @CTRL, DESC : 0528
                                ADDL3 #1, CTRL, DESC+4 : 0529
                                PUSHAB PARAM : 0530
                                PUSHL R2 :
                                PUSHAB OUTLEN :
                                PUSHAB DESC :

```

DUMPSHEADER
V04-000

B 1
16-Sep-1984 01:30:28
14-Sep-1984 12:21:37

VAX-11 Bliss-32 V4.0-742
[DUMP.SRC]DUMPHEADE.B32;1

Page 14
(5)

00000000G	00	04	FB	00022	CALLS	#4, SYSS\$AOL	:
	50	6E	3C	00029	MOVZWL	OUTLEN, RO	: 0531
	62	50	C2	0002C	SUBL2	RO, LINE_DESC	:
	50	6E	3C	0002F	MOVZWL	OUTLEN, RO	: 0532
04	A2	50	C0	00032	ADDL2	RO, LINE_DESC+4	:
		04	00036	RET			: 0533

; Routine Size: 55 bytes. Routine Base: \$CODE\$ + 01D4


```

: 424 0534 1 ROUTINE DUMP$EOL: NOVALUE=
: 425 0535 1
: 426 0536 1 ++
: 427 0537 1
: 428 0538 1 FUNCTIONAL DESCRIPTION:
: 429 0539 1 This routine writes the contents of the line buffer and reinitializes
: 430 0540 1 for a new line.
: 431 0541 1
: 432 0542 1 INPUT PARAMETERS:
: 433 0543 1 NONE
: 434 0544 1
: 435 0545 1 IMPLICIT INPUTS:
: 436 0546 1 LINE_DESC - Descriptor for line.
: 437 0547 1
: 438 0548 1 OUTPUT PARAMETERS:
: 439 0549 1 NONE
: 440 0550 1
: 441 0551 1 IMPLICIT OUTPUTS:
: 442 0552 1 NONE
: 443 0553 1
: 444 0554 1 ROUTINE VALUE:
: 445 0555 1 NONE
: 446 0556 1
: 447 0557 1 SIDE EFFECTS:
: 448 0558 1 NONE
: 449 0559 1
: 450 0560 1 --
: 451 0561 1
: 452 0562 2 BEGIN
: 453 0563 2 LINE_DESC[0] = .DUMP$GL_WIDTH - .LINE_DESC[0];
: 454 0564 2 LINE_DESC[1] = .DUMP$GL_OUTDESC[DSC$A_POINTER];
: 455 0565 2 DUMP$PUT_LINE(LINE_DESC);
: 456 0566 2 LINE_DESC[0] = .DUMP$GL_WIDTH;
: 457 0567 1 END;

```

```

                                000C 0000 DUMP$EOL:
                                .WORD Save R2,R3 : 0534
                                53 00000000G 00 9E 00002 MOVAB DUMP$GL_WIDTH, R3
                                52 00000000' EF 9E 00009 MOVAB LINE_DESC, R2
62                                63 00000000G 62 C3 00010 SUBL3 LINE_DESC, DUMP$GL_WIDTH, LINE_DESC : 0563
                                04 A2 00000000G 00 D0 00014 MOVL DUMP$GL_OUTDESC+4, LINE_DESC+4 : 0564
                                52 DD 0001C PUSHL R2 : 0565
                                00000000G 00 01 FB 0001E CALLS #1, DUMP$PUT_LINE
                                62 63 D0 00025 MOVL DUMP$GL_WIDTH, LINE_DESC : 0566
                                04 00028 RET : 0567

```

; Routine Size: 41 bytes, Routine Base: \$CODE\$ + 020B

```

: 459      0568 1 ROUTINE DUMP$PUT_FAO_(FAO,PARAM): NOVALUE=
: 460      0569 1
: 461      0570 1 :++
: 462      0571 1
: 463      0572 1 FUNCTIONAL DESCRIPTION:
: 464      0573 1 This routine combines the actions of DUMP$FAO_ and DUMP$EOL.
: 465      0574 1
: 466      0575 1 INPUT PARAMETERS:
: 467      0576 1 As for DUMP$FAO_.
: 468      0577 1
: 469      0578 1 IMPLICIT INPUTS:
: 470      0579 1 NONE
: 471      0580 1
: 472      0581 1 OUTPUT PARAMETERS:
: 473      0582 1 NONE
: 474      0583 1
: 475      0584 1 IMPLICIT OUTPUTS:
: 476      0585 1 NONE
: 477      0586 1
: 478      0587 1 ROUTINE VALUE:
: 479      0588 1 NONE
: 480      0589 1
: 481      0590 1 SIDE EFFECTS:
: 482      0591 1 NONE
: 483      0592 1
: 484      0593 1 --
: 485      0594 1
: 486      0595 2 BEGIN
: 487      0596 2 BUILTIN
: 488      0597 2 CALLG,
: 489      0598 2 AP;
: 490      0599 2
: 491      0600 2
: 492      0601 2 CALLG(.AP, DUMP$FAO_);
: 493      0602 2 DUMP$EOL();
: 494      0603 1 END;

```

		0000 0000	DUMP\$PUT_FAO :		
			WORD	Save nothing	: 0568
9A	AF	6C FA 0002	CALLG	(AP), DUMP\$FAO_	: 0601
CD	AF	00 FB 0006	CALLS	#0, DUMP\$EOL	: 0602
		04 0000A	RET		: 0603

: Routine Size: 11 bytes. Routine Base: \$CODE\$ + 0234

```

: 496 0604 1 ROUTINE FORMAT_BIT_STRING(VALUE, CODES): NOVALUE=
: 497 0605 1
: 498 0606 1 ++
: 499 0607 1
: 500 0608 1 FUNCTIONAL DESCRIPTION:
: 501 0609 1 This routine formats a bit mask.
: 502 0610 1
: 503 0611 1 INPUT PARAMETERS:
: 504 0612 1 VALUE - The bit mask value.
: 505 0613 1 CODES - Pointer to a PLIT of ASCII bit names.
: 506 0614 1
: 507 0615 1 IMPLICIT INPUTS:
: 508 0616 1 NONE
: 509 0617 1
: 510 0618 1 OUTPUT PARAMETERS:
: 511 0619 1 NONE
: 512 0620 1
: 513 0621 1 IMPLICIT OUTPUTS:
: 514 0622 1 NONE
: 515 0623 1
: 516 0624 1 ROUTINE VALUE:
: 517 0625 1 NONE
: 518 0626 1
: 519 0627 1 SIDE EFFECTS:
: 520 0628 1 The listing is produced.
: 521 0629 1
: 522 0630 1 --
: 523 0631 1
: 524 0632 2 BEGIN
: 525 0633 2 MAP
: 526 0634 2 CODES: REF VECTOR; ! Value names
: 527 0635 2 LOCAL
: 528 0636 2 INIT_USED, ! Initial used space on line
: 529 0637 2 V; ! Local copy of value
: 530 0638 2
: 531 0639 2
: 532 0640 2 INIT_USED = .DUMPSGL_WIDTH - .LINE_DESC[0];
: 533 0641 2 V = .VALUE;
: 534 0642 2 IF .V EQL 0
: 535 0643 2 THEN
: 536 0644 2 DUMPSFA0('<none specified>')
: 537 0645 2 ELSE
: 538 0646 2 INCR N FROM 0 TO 31 DO
: 539 0647 3 BEGIN
: 540 0648 3 IF .V
: 541 0649 3 THEN
: 542 0650 4 BEGIN
: 543 0651 5 IF (IF .N GEQU .CODES[-1] THEN TRUE ELSE .CODES[.N] EQL 0)
: 544 0652 4 THEN
: 545 0653 5 BEGIN
: 546 0654 5 IF .LINE_DESC[0] LSSU 8
: 547 0655 5 THEN
: 548 0656 6 BEGIN
: 549 0657 6 DUMPEOL();
: 550 0658 6 DUMPSFA0('!#* ', .INIT_USED);
: 551 0659 5 END;
: 552 0660 5 DUMPSFA0('Bit !UL', .N);

```

```

: 553      0661  5      END
: 554      0662  4      ELSE
: 555      0663  3      BEGIN
: 556      0664  3      IF .LINE_DESC[0] LSSU 2 + .VECTOR[.CODES[N], 0; ,BYTE]
: 557      0665  5      THEN
: 558      0666  6      BEGIN
: 559      0667  6      DUMP$EOL();
: 560      0668  6      DUMP$FAO('!#* ', .INIT_USED);
: 561      0669  5      END;
: 562      0670  5      DUMP$FAO('!AC', .CODES[N]);
: 563      0671  4      END;
: 564      0672  4      V = .V<1,31>;
: 565      0673  4      IF .V EQL 0 THEN EXITLOOP;
: 566      0674  4      DUMP$FAO(' ', ');
: 567      0675  4      END
: 568      0676  3      ELSE
: 569      0677  3      V = .V<1,31>;
: 570      0678  2      END;
: 571      0679  2      DUMP$EOL();
: 572      0680  1      END;

```

```

                                .PSECT $PLITS$,NOWRT,NOEXE,2
65 69 66 69 63 65 70 73 20 65 6E 6F 6E 3C 10 00000 P.AAA: .ASCII <16>\<none specified>\
                                3E 64 0000F
                                20 2A 23 21 04 00011 P.AAB: .ASCII <4>\!#* \
                                4C 55 21 20 74 69 42 07 00016 P.AAC: .ASCII <7>\Bit !UL\
                                20 2A 23 21 04 0001E P.AAD: .ASCII <4>\!#* \
                                43 41 21 03 00023 P.AAE: .ASCII <3>\!AC\
                                20 2C 02 00027 P.AAF: .ASCII <2>\, \

```

```

                                .PSECT $CODE$,NOWRT,2
                                01FC 00000 FORMAT_BIT STRING:
                                .WORD Save R2,R3,R4,R5,R6,R7,R8 : 0604
58 00000000' EF 9E 00002 MOVAB LINE_DESC, R8
57 00000000' EF 9E 00009 MOVAB P.AAA, R7
56 82 AF 9E 00010 MOVAB DUMP$FAO, R6
55 00000000G 00 68 C3 00014 SUBL3 LINE_DESC, DUMP$GL_WIDTH, INIT_USED : 0640
54 04 AC D0 0001C MOVL VALUE, V : 0641
07 12 00020 BNEQ 1$ : 0642
57 DD 00022 PUSHL R7 : 0644
66 01 FB 00024 CALLS #1, DUMP$FAO_
67 11 00027 BRB 10$
53 D4 00029 1$: CLRL N : 0646
59 54 E9 0002B 2$: BLBC V, 8$ : 0648
52 08 AC D0 0002E MOVL CODES, R2 : 0651
FC A2 53 D1 00032 CMPL N, -4(R2)
05 1E 00036 BGEQU 3$
6243 D5 00038 TSTL (R2)[N]
18 12 0003B BNEQ 5$
08 68 D1 0003D 3$: CMPL LINE_DESC, #8 : 0654
0C 1E 00040 BGEQU 4$

```

37	A6	00	FB	00042	CALLS	#0, DUMPSEOL	:	0657
		55	DD	00046	PUSHL	INIT_USED	:	0658
		11	A7	9F 00048	PUSHAB	P.AAB	:	
	66	02	FB	0004B	CALLS	#2, DUMPSFAO_	:	
		53	DD	0004E 4\$:	PUSHL	N	:	0660
		16	A7	9F 00050	PUSHAB	P.AAC	:	
		20	11	00053	BRB	7\$:	
	52	6243	DD	00055 5\$:	MOVL	(R2)[N], R2	:	0664
	50	62	9A	00059	MOVZBL	(R2), R0	:	
	50	02	CO	0005C	ADDL2	#2, R0	:	
	50	68	D1	0005F	CMPL	LINE_DESC, R0	:	
		0C	1E	00062	BGEQU	6\$:	
	37	A6	00	FB 00064	CALLS	#0, DUMPSEOL	:	0667
			55	DD 00068	PUSHL	INIT_USED	:	0668
		1E	A7	9F 0006A	PUSHAB	P.AAB	:	
	66		02	FB 0006D	CALLS	#2, DUMPSFAO_	:	
			52	DD 00070 6\$:	PUSHL	R2	:	0670
		23	A7	9F 00072	PUSHAB	P.AAE	:	
	66		02	FB 00075 7\$:	CALLS	#2, DUMPSFAO_	:	
54		1F	01	EF 00078	EXTZV	#1, #31, V, V	:	0672
			11	13 0007D	BEQL	10\$:	0673
		27	A7	9F 0007F	PUSHAB	P.AAF	:	0674
	66		01	FB 00082	CALLS	#1, DUMPSFAO_	:	
			05	11 00085	BRB	9\$:	0648
54		1F	01	EF 00087 8\$:	EXTZV	#1, #31, V, V	:	0677
		53	1F	F3 0008C 9\$:	AOBLEQ	#31, N, 2\$:	0646
	37	A6	00	FB 00090 10\$:	CALLS	#0, DUMPSEOL	:	0679
			04	00094	RET		:	0680

: Routine Size: 149 bytes, Routine Base: \$CODE\$ + 023F

```

: 574 0681 1 ROUTINE FORMAT_CODED_VALUE(VALUE, CODES): NOVALUE=
: 575 0682 1
: 576 0683 1 !++
: 577 0684 1
: 578 0685 1 FUNCTIONAL DESCRIPTION:
: 579 0686 1 This routine formats a coded integer value.
: 580 0687 1
: 581 0688 1 INPUT PARAMETERS:
: 582 0689 1 VALUE - The coded value.
: 583 0690 1 CODES - Pointer to a PLIT of ASCII value names.
: 584 0691 1
: 585 0692 1 IMPLICIT INPUTS:
: 586 0693 1 NONE
: 587 0694 1
: 588 0695 1 OUTPUT PARAMETERS:
: 589 0696 1 NONE
: 590 0697 1
: 591 0698 1 IMPLICIT OUTPUTS:
: 592 0699 1 NONE
: 593 0700 1
: 594 0701 1 ROUTINE VALUE:
: 595 0702 1 NONE
: 596 0703 1
: 597 0704 1 SIDE EFFECTS:
: 598 0705 1 The listing is produced.
: 599 0706 1
: 600 0707 1 !--
: 601 0708 1
: 602 0709 2 BEGIN
: 603 0710 2 MAP
: 604 0711 2 CODES: REF VECTOR; ! Value names
: 605 0712 2
: 606 0713 2
: 607 0714 2 IF .VALUE GEQU .CODES[-1]
: 608 0715 2 THEN
: 609 0716 2 DUMP$PUT_FAO('!UL', .VALUE)
: 610 0717 2 ELSE
: 611 0718 2 DUMP$PUT_FAO('!AC', .CODES[.VALUE]);
: 612 0719 1 END;

```

.PSECT \$PLITS, NOWRT, NOEXE, 2

```

4C 55 21 03 0002A P.AAG: .ASCII <3>\!UL\
43 41 21 03 0002E P.AAH: .ASCII <3>\!AC\

```

.PSECT \$CODES, NOWRT, 2

```

000C 0000 FORMAT_CODED_VALUE:
FC 52 04 AC 7D 00002 .WORD Save R2, R3
A3 52 D1 00006 MOVQ VALUE, R2
0A 1F 0000A CMPL R2, -4(R3)
52 DD 0000C BLSSU 1$
PUSHL R2

```

```

: 0681
: 0714
:
: 0716

```

DUMPSHEADER
V04-000

I 1
16-Sep-1984 01:30:28
14-Sep-1984 12:21:37

VAX-11 Bliss-32 V4.0-742
[DUMP.SRC]DUMPHEADE.B32;1

Page 21
(9)

	00000000'	EF	9F	0000E		PUSHAB	P.AAG	
		09	11	00014		BRB	2\$	
		6342	DD	00016	1\$:	PUSHL	(R3)[R2]	0718
	00000000'	EF	9F	00019		PUSHAB	P.AAH	
FF3C	CF	02	FB	0001F	2\$:	CALLS	#2, DUMPSPUT_FAO_	
			04	00024		RET		0719

; Routine Size: 37 bytes, Routine Base: \$CODE\$ + 02D4

```

: 614 0720 1 ROUTINE FORMAT_DATE_VALUE(VALUE): NOVALUE=
: 615 0721 1
: 616 0722 1 |++
: 617 0723 1
: 618 0724 1 | FUNCTIONAL DESCRIPTION:
: 619 0725 1 | This routine formats a standard system date.
: 620 0726 1
: 621 0727 1 | INPUT PARAMETERS:
: 622 0728 1 | VALUE - Pointer to the system date.
: 623 0729 1
: 624 0730 1 | IMPLICIT INPUTS:
: 625 0731 1 | NONE
: 626 0732 1
: 627 0733 1 | OUTPUT PARAMETERS:
: 628 0734 1 | NONE
: 629 0735 1
: 630 0736 1 | IMPLICIT OUTPUTS:
: 631 0737 1 | NONE
: 632 0738 1
: 633 0739 1 | ROUTINE VALUE:
: 634 0740 1 | NONE
: 635 0741 1
: 636 0742 1 | SIDE EFFECTS:
: 637 0743 1 | The listing is produced.
: 638 0744 1
: 639 0745 1 | --
: 640 0746 1
: 641 0747 2 BEGIN
: 642 0748 2 MAP
: 643 0749 2 VALUE: REF VECTOR; ! Pointer to date
: 644 0750 2
: 645 0751 2
: 646 0752 2 IF .VALUE[0] EQL 0
: 647 0753 2 THEN
: 648 0754 2 DUMP$PUT_FA0('<none specified>')
: 649 0755 2 ELSE
: 650 0756 2 DUMP$PUT_FA0('!%D', .VALUE);
: 651 0757 2 END;

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2
65 69 66 69 63 65 70 73 20 65 6E 6F 6E 3C 10 00032 P.AAI: .ASCII <16>\<none specified>\
: 3E 64 00041
: 44 25 21 03 00043 P.AAJ: .ASCII <3>\!%D\

```

```

.PSECT $CODE$,NOWRT,2
0004 00000 FORMAT_DATE_VALUE:
: 0720
52 FF35 CF 9E 00002 .WORD Save R2
: 0752
: 04 BC D5 00007 MOVAB DUMP$PUT_FA0_, R2
: TSTL @VALUE
: BNEQ 1$
: 0754
00000000' EF 9F 0000C PUSHAB P.AAI

```



```

: 653      0758 1 ROUTINE FORMAT_ODS1_DATE_VALUE(VALUE): NOVALUE=
: 654      0759 1
: 655      0760 1 !++
: 656      0761 1
: 657      0762 1 FUNCTIONAL DESCRIPTION:
: 658      0763 1     This routine formats an ODS-1 date.
: 659      0764 1
: 660      0765 1 INPUT PARAMETERS:
: 661      0766 1     VALUE           - Pointer to the ODS-1 date.
: 662      0767 1
: 663      0768 1 IMPLICIT INPUTS:
: 664      0769 1     NONE
: 665      0770 1
: 666      0771 1 OUTPUT PARAMETERS:
: 667      0772 1     NONE
: 668      0773 1
: 669      0774 1 IMPLICIT OUTPUTS:
: 670      0775 1     NONE
: 671      0776 1
: 672      0777 1 ROUTINE VALUE:
: 673      0778 1     NONE
: 674      0779 1
: 675      0780 1 SIDE EFFECTS:
: 676      0781 1     The listing is produced.
: 677      0782 1
: 678      0783 1 --
: 679      0784 1
: 680      0785 2 BEGIN
: 681      0786 2 MAP
: 682      0787 2     VALUE:           REF VECTOR[,BYTE];     ! Pointer to date
: 683      0788 2
: 684      0789 2
: 685      0790 2 IF .VALUE[0] EQL 0
: 686      0791 2 THEN
: 687      0792 2     DUMPSFAO('<none specified>')
: 688      0793 2 ELSE
: 689      0794 2     BEGIN
: 690      P 0795 3     DUMPSFAO('!AF-!AF-19!AF',
: 691      P 0796 3         2, VALUE[0],
: 692      P 0797 3         3, VALUE[2],
: 693      0798 3         2, VALUE[5]);
: 694      0799 3     IF .VALUE[7] NEQ 0
: 695      0800 3     THEN
: 696      P 0801 3         DUMPSFAO('!AF:!AF:!AF',
: 697      P 0802 3             2, VALUE[7],
: 698      P 0803 3             2, VALUE[9],
: 699      0804 3             2, VALUE[11]);
: 700      0805 2     END;
: 701      0806 2 DUMPS_EOL();
: 702      0807 1 END;

```

.PSECT \$SPLITS,NOWRT,NOEXE,2

65 69 66 69 63 65 70 73 20 65 6E 6F 6E 3C 10 00047 P.AAK: .ASCII <16>\<none specified>\
3E 64 00056

46 41 21 39 31 2D 46 41 21 2D 46 41 21 0D 00058 P.AAL: .ASCII <13>\!AF-!AF-19!AF\
46 41 21 3A 46 41 21 3A 46 41 21 0C 00066 P.AAM: .ASCII <12>\.AF:!AF:!AF\
:

.PSECT \$CODE\$,NOWRT,2

```

001C 00000 FORMAT_ODS1 DATE VALUE:
54 00000000' EF 9E 00002 .WORD Save R2,R3,R4 : 0758
53 FEAB CF 9E 00009 MOVAB P.AAK, R4
52 04 AC 0D 0000E MOVAB DUMPSFAO, R3 : 0790
62 95 00012 TSTB (R2)
07 12 00014 BNEQ 1$ : 0792
54 DD 00016 PUSHL R4 : 0798
63 01 FB 00018 CALLS #1, DUMPSFAO_
2E 11 00018 BRB 2$
05 A2 9F 0001D 1$: PUSHAB 5(R2) : 0798
02 A2 9F 00022 PUSHL #2
03 DD 00025 PUSHAB 2(R2)
52 DD 00027 PUSHL #3
02 DD 00029 PUSHL R2
63 11 A4 9F 0002B PUSHL #2
07 FB 0002E CALLS #7, DUMPSFAO_
07 A2 95 00031 TSTB 7(R2) : 0799
15 13 00034 BEQL 2$
08 A2 9F 00036 PUSHAB 11(R2) : 0804
02 DD 00039 PUSHL #2
09 A2 9F 0003B PUSHAB 9(R2)
02 DD 0003E PUSHL #2
07 A2 9F 00040 PUSHAB 7(R2)
02 DD 00043 PUSHL #2
1F A4 9F 00045 PUSHAB P.AAM
37 63 07 FB 00048 CALLS #7, DUMPSFAO_
A3 00 FB 0004B 2$: CALLS #0, DUMPSEOL : 0806
04 0004F RET : 0807

```

; Routine Size: 80 bytes, Routine Base: \$CODE\$ + 031C

```

: 704      0808 1 ROUTINE FORMAT_PROTECTION(PROT,CODES): NOVALUE=
: 705      0809 1
: 706      0810 1 !++
: 707      0811 1
: 708      0812 1 FUNCTIONAL DESCRIPTION:
: 709      0813 1     This routine formats a protection value.
: 710      0814 1
: 711      0815 1 INPUT PARAMETERS:
: 712      0816 1     PROT           - A protection value.
: 713      0817 1     CODES          - The four letter codes, e.g. 'RWED'.
: 714      0818 1
: 715      0819 1 IMPLICIT INPUTS:
: 716      0820 1     NONE
: 717      0821 1
: 718      0822 1 OUTPUT PARAMETERS:
: 719      0823 1     NONE
: 720      0824 1
: 721      0825 1 IMPLICIT OUTPUTS:
: 722      0826 1     NONE
: 723      0827 1
: 724      0828 1 ROUTINE VALUE:
: 725      0829 1     NONE
: 726      0830 1
: 727      0831 1 SIDE EFFECTS:
: 728      0832 1     The listing is produced.
: 729      0833 1
: 730      0834 1 --
: 731      0835 1
: 732      0836 2 BEGIN
: 733      0837 2 MAP
: 734      0838 2     PROT:           BITVECTOR[16], ! Protection value
: 735      0839 2     CODES:          VECTOR[BYTE]; ! Letter codes
: 736      0840 2 BIND
: 737      0841 2     WHO = UPLIT (
: 738      0842 2         UPLIT BYTE (%ASCIC 'S:'),
: 739      0843 2         UPLIT BYTE (%ASCIC ' ', 0:'),
: 740      0844 2         UPLIT BYTE (%ASCIC ' ', G:'),
: 741      0845 2         UPLIT BYTE (%ASCIC ' ', W:'))
: 742      0846 2     : VECTOR;
: 743      0847 2
: 744      0848 2
: 745      0849 2 INCR I FROM 0 TO 3 DO
: 746      0850 3     BEGIN
: 747      0851 3     DUMPSFAO('!AC', .WHO[I]);
: 748      0852 3     INCR J FROM 0 TO 3 DO
: 749      0853 4         BEGIN
: 750      0854 4         IF NOT .PROT[I*4+.J] THEN DUMPSFAO('!AD', 1, CODES[J]);
: 751      0855 3         END;
: 752      0856 2     END;
: 753      0857 2
: 754      0858 2
: 755      0859 2 DUMPS_EOL();
: 756      0860 1 END;

```

.PSECT SPLITS,NOWRT,NOEXE,2

```

3A 4F 20 2C 04 00073 P.AAO: .ASCII <2>\S:\
3A 47 20 2C 04 00076 P.AAP: .ASCII <4>\, O:\
3A 57 20 2C 04 00078 P.AAQ: .ASCII <4>\, G:\
3A 57 20 2C 04 00080 P.AAR: .ASCII <4>\, W:\
00085 .BLKB 3
00000000' 00000000' 00000000' 00000000' 00088 P.AAN: .ADDRESS P.AAO, P.AAP, P.AAQ, P.AAR
43 41 21 03 00098 P.AAS: .ASCII <3>\!AC\
44 41 21 03 0009C P.AAT: .ASCII <3>\!AD\

```

WHO= P.AAN

.PSECT \$CODE\$,NOWRT,2

```

003C 00000 FORMAT_PROTECTION:
55 00000000' EF 9E 00002 .WORD Save R2,R3,R4,R5 : 0808
54 FE5B CF 9E 00009 MOVAB WHO, R5 :
53 D4 0000E CLRL I : 0849
6543 DD 00010 1$: PUSHL WHO[I] : 0851
10 A5 9F 00013 PUSHAB P.AAS :
64 02 FB 00016 CALLS #2, DUMPSFAO_ :
52 D4 00019 CLRL J : 0852
50 6243 DE 0001B 2$: MOVAL (J)[I], R0 : 0854
OC 04 AC 50 E0 0001F BBS R0, PROT, 3$ :
08 AC42 9F 00024 PUSHAB CODES[J] :
01 DD 00028 PUSHL #1 :
14 A5 9F 0002A PUSHAB P.AAT :
64 03 FB 0002D CALLS #3, DUMPSFAO_ :
E7 52 03 F3 00030 3$: AOBLEQ #3, J, 2$ : 0852
DB 53 03 F3 00034 AOBLEQ #3, I, 1$ : 0849
37 A4 00 FB 00038 CALLS #0, DUMPEOL : 0859
04 0003C RET : 0860

```

; Routine Size: 61 bytes, Routine Base: \$CODE\$ + 036C

```
0861 1 ROUTINE FORMAT_RMS_ATTRIBUTES(FAT): NOVALUE=  
0862 1  
0863 1 !++  
0864 1  
0865 1 FUNCTIONAL DESCRIPTION:  
0866 1 This routine formats the RMS attributes.  
0867 1  
0868 1 INPUT PARAMETERS:  
0869 1 FAT - Pointer to attributes area.  
0870 1  
0871 1 IMPLICIT INPUTS:  
0872 1 NONE  
0873 1  
0874 1 OUTPUT PARAMETERS:  
0875 1 NONE  
0876 1  
0877 1 IMPLICIT OUTPUTS:  
0878 1 NONE  
0879 1  
0880 1 ROUTINE VALUE:  
0881 1 NONE  
0882 1  
0883 1 SIDE EFFECTS:  
0884 1 NONE  
0885 1  
0886 1 --  
0887 1  
0888 2 BEGIN  
0889 2 MAP  
0890 2 FAT: REF BBLOCK; ! Pointer to RMS attributes area  
0891 2 BUILTIN  
0892 2 ROT;  
0893 2 BIND  
0894 2 RTYPE CODES = PLIT(  
0895 2 UPLIT BYTE (%ASCIC 'Undefined'),  
0896 2 UPLIT BYTE (%ASCIC 'Fixed'),  
0897 2 UPLIT BYTE (%ASCIC 'Variable'),  
0898 2 UPLIT BYTE (%ASCIC 'Variable with fixed control'),  
0899 2 UPLIT BYTE (%ASCIC 'RMS-11 stream'),  
900 2 UPLIT BYTE (%ASCIC 'Lf-terminated stream'),  
901 2 UPLIT BYTE (%ASCIC 'CR-terminated stream')),  
902 2 FILEORG CODES = PLIT(  
903 2 UPLIT BYTE (%ASCIC 'Sequential'),  
904 2 UPLIT BYTE (%ASCIC 'Relative'),  
905 2 UPLIT BYTE (%ASCIC 'Indexed'),  
906 2 UPLIT BYTE (%ASCIC 'Direct')),  
907 2 RATTRIB CODES = PLIT(  
908 2 UPLIT BYTE (%ASCIC 'FORTRAN carriage control'),  
909 2 UPLIT BYTE (%ASCIC 'Implied carriage control'),  
910 2 UPLIT BYTE (%ASCIC 'Print carriage control'),  
911 2 UPLIT BYTE (%ASCIC 'Non-spanned'));  
912 2  
913 2  
914 2 DUMPSPUT LINE($DESCRIPTOR(' VAX-11 RMS attributes'));  
915 2 DUMPSFAOT '!8* Record type:!22* ');  
916 2 FORMAT CODED VALUE(.FAT[FAT&V RTYPE], RTYPE_CODES);  
917 2 DUMPSFAO('!8* File organization:!T6* ');
```

```

: 815      0918 2    FORMAT CODED VALUE(.FAT[FAT$V FILEORG], FILEORG_CODES);
: 816      0919 2    DUMPSFAO('T8* Record attributes:'T6* ');
: 817      0920 2    FORMAT BIT STRING(.FAT[FAT$B RATTRIB], RATTRIB_CODES);
: 818      P 0921 2    DUMPSPUT FAO('T8* Record size:.22* !UL',
: 819      0922 2    .FAT[FAT$W RSIZE]);
: 820      P 0923 2    DUMPSPUT FAO('T8* Highest block:!20* !UL',
: 821      0924 2    ROT(.FAT[FAT$L HIBLK], 16));
: 822      P 0925 2    DUMPSPUT FAO('!8* End of file block:!16* !UL',
: 823      0926 2    ROT(.FAT[FAT$L EFBLK], 16));
: 824      P 0927 2    DUMPSPUT FAO('!8* End of file byte:!17* !UL',
: 825      0928 2    .FAT[FAT$W FFBYTE]);
: 826      P 0929 2    DUMPSPUT FAO('T8* Bucket size:!22* !UL',
: 827      0930 2    .FAT[FAT$B BKTSIZE]);
: 828      P 0931 2    DUMPSPUT FAO('T8* Fixed control area size:!10* !UL',
: 829      0932 2    .FAT[FAT$B VFCSIZE]);
: 830      P 0933 2    DUMPSPUT FAO('T8* Maximum record size:!14* !UL',
: 831      0934 2    .FAT[FAT$W MAXREC]);
: 832      P 0935 2    DUMPSPUT FAO('T8* Default extension size:!11* !UL',
: 833      0936 2    .FAT[FAT$W DEFEXT]);
: 834      P 0937 2    DUMPSPUT FAO('T8* Global buffer count:!14* !UL',
: 835      0938 2    .FAT[FAT$W GBC]);
: 836      P 0939 2    DUMPSPUT FAO('T8* Directory version limit:!10* !UL',
: 837      0940 2    .FAT[FAT$W_VERSIONS]);
: 838      0941 1    END;

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2

      64 65 6E 69 66 65 64 6E 55 09 000A0 P.AAV: .ASCII <9>\Undefined\
      64 65 78 69 46 05 000AA P.AAW: .ASCII <5>\Fixed\
20 68 74 69 77 20 65 6C 62 61 69 72 61 56 08 000B0 P.AAX: .ASCII <8>\Variable\
      6C 6F 72 74 6E 6F 63 20 64 65 78 69 66 000C8 P.AAY: .ASCII <27>\Variable with fixed control\
20 6D 61 65 72 74 73 20 31 31 2D 53 4D 52 0D 000D5 P.AAZ: .ASCII <13>\RMS-11 stream\
      64 65 74 61 6E 69 6D 72 65 74 2D 46 4C 14 000E3 P.ABA: .ASCII <20>\LF-terminated stream\
      6D 61 65 72 74 73 000F2
20 64 65 74 61 6E 69 6D 72 65 74 2D 52 43 14 000F8 P.ABB: .ASCII <20>\CR-terminated stream\
      6D 61 65 72 74 73 00107
      0010D .BLKB 3
      00110 .LONG 7
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00114 P.AAU: .ADDRESS P.AAV, P.AAW, P.AAX, P.AAY, P.AAZ, -
      0G000000' 0012C P.AAB: .ADDRESS P.ABA, P.ABB
      6C 61 69 74 6E 65 75 71 65 53 0A 00130 P.ABD: .ASCII <10>\Sequential\
      65 76 69 74 61 6C 65 52 08 0013B P.ABE: .ASCII <8>\Relative\
      64 65 78 65 64 6E 49 07 00144 P.ABF: .ASCII <7>\Indexed\
      74 63 65 72 69 44 06 0014C P.ABG: .ASCII <6>\Direct\
      00153 .BLKB 1
      00000004' 00154 .LONG 4
61 69 72 72 61 63 20 4E 41 52 54 52 4F 46 18 00158 P.ABC: .ADDRESS P.ABD, P.ABE, P.ABF, P.ABG
      6C 6F 72 74 6E 6F 63 2D 65 67 00168 P.ABI: .ASCII <24>\FORTRAN carriage control\
61 69 72 72 61 63 20 64 65 69 6C 70 6D 49 18 00177
      6C 6F 72 74 6E 6F 63 2D 65 67 00181 P.ABJ: .ASCII <24>\Implied carriage control\
65 67 61 69 72 72 61 63 20 74 6E 69 72 50 16 00190 P.ABK: .ASCII <22>\Print carriage control\
      6C 6F 72 74 6E 6F 63 20 001A9
      64 65 6E 6E 61 70 73 2D 6E 6F 4E 0B 001B1 P.ABL: .ASCII <11>\Non-spanned\

```

```

00000000' 00000000' 00000000' 00000004'
31 31 2D 58 41 56 20 20 20 20 20
73 65 74 75 62 69 72 74 74 61
001BD .BLKB 3
001C0 .LONG 4
001C4 P.ABH: .ADDRESS P.ABI, P.ABJ, P.ABK, P.ABL
001D4 P.ABN: .ASCII \ VAX-11 RMS attributes\
001E3
001ED .BLKB 3
001F0 P.ABM: .LONG 25
001F4 .ADDRESS P.ABN
001F8 P.ABO: .ASCII <21>\!8* Record type:!22* \
0207
020E P.ABP: .ASCII <27>\!8* File organization:!16* \
021D
022A P.ABQ: .ASCII <27>\!8* Record attributes:!16* \
0239
0246 P.ABR: .ASCII <24>\!8* Record size:!22* !UL\
0255
025F P.ABS: .ASCII <26>\!8* Highest block:!20* !UL\
026E
027A P.ABT: .ASCII <30>\!8* End of file block:!16* !UL\
0289
0298
0299 P.ABU: .ASCII <29>\!8* End of file byte:!17* !UL\
02A8
02B7 P.ABV: .ASCII <24>\!8* Bucket size:!22* !UL\
02C6
02D0 P.ABW: .ASCII \$!8* Fixed control area size:!10* !UL\
02DF
02EE
02F5 P.ABX: .ASCII \!8* Maximum record size:!14* !UL\
0304
0313
0316 P.ABY: .ASCII \#!8* Default extension size:!11* !UL\
0325
0334
033A P.ABZ: .ASCII \!8* Global buffer count:!14* !UL\
0349
0358
035B P.ACA: .ASCII \$!8* Directory version limit:!10* !UL\
036A
0379

```

```

RTYPE_CODES= P.AAU
FILEORG_CODES= P.ABC
RATTRIB_CODES= P.ABH

```

.PSECT \$CODE\$,NOWRT,2

```

001C 0000 FORMAT_RMS_ATTRIBUTES:
54 FE85 CF 9E 00002 .WORD Save R2,R3,R4 : 0861
53 00000000' EF 9E 00007 MOVAB DUMPSPUT_FAO_, R4
00000000G 00 53 DD 0000E MOVAB P.ABM, R3
A0 A4 08 01 FB 00010 PUSHL R3 : 0914
FF24 C3 9F 00017 CALLS #1, DUMPSPUT_LINE
01 FB 0001A PUSHAB P.ABO : 0915
01 FB 0001E CALLS #1, DUMPSFAG_
PUSHAB RTYPE_CODES : 0916

```


7E	62	00A0	52 04 C4	04	AC 00 02	DO EF FB	00022 00026 0002B	MOVL EXTZV CALLS	FAT, R2 #0, #4, (R2), -(SP) #2, FORMAT_CODED_VALUE	:	
		A0	A4	1E	A3	9F	00030	PUSHAB	P.ABP	:	0917
				FF68	01	FB	00033	CALLS	#1, DUMPSFAO	:	
7E	62	00A0	04 C4		C3	9F	00037	PUSHAB	FILEORG_CODES	:	0918
		A0	A4	3A	02	FB	00040	CALLS	#4, #4, (R2), -(SP) #2, FORMAT_CODED_VALUE	:	
					A3	9F	00045	PUSHAB	P.ABQ	:	0919
		A0	A4	D4	01	FB	00048	CALLS	#1, DUMPSFAO	:	
			7E	01	A3	9F	0004C	PUSHAB	RATTRIB_CODES	:	0920
		0B	A4		A2	9A	0004F	MOVZBL	1(R2), -(SP)	:	
			7E	02	A2	FB	00053	CALLS	#2, FORMAT_BIT_STRING	:	
			7E	56	A3	9F	0005B	MOVZWL	2(R2), -(SP)	:	0922
			64		A2	FB	0005E	PUSHAB	P.ABR	:	
7E	04	A2	A2		02	FB	00061	CALLS	#2, DUMPSPUT_FAO	:	
				6F	10	9C	00061	ROTL	#16, 4(R2), =(SPT)	:	0924
			64		A3	9F	00066	PUSHAB	P.ABS	:	
		7E	A2		02	FB	00069	CALLS	#2, DUMPSPUT_FAO	:	
			7E	08	10	9C	0006C	ROTL	#16, 8(R2), =(SPT)	:	0926
			64	008A	C3	9F	00071	PUSHAB	P.ABT	:	
			7E	0C	A2	FB	00075	CALLS	#2, DUMPSPUT_FAO_	:	
			64	00A9	C3	3C	00078	MOVZWL	12(R2), -(SP)	:	0928
			7E	0E	A2	9F	0007C	PUSHAB	P.ABU	:	
			64	00C7	02	FB	00080	CALLS	#2, DUMPSPUT_FAO_	:	
			7E		A2	9A	00083	MOVZBL	14(R2), -(SP)	:	0930
			64		C3	9F	00087	PUSHAB	P.ABV	:	
			7E	0F	A2	FB	0008B	CALLS	#2, DUMPSPUT_FAO_	:	
			7E	00E0	C3	9A	0008E	MOVZBL	15(R2), -(SP)	:	0932
			64		A2	9F	00092	PUSHAB	P.ABW	:	
			7E	10	02	FB	00096	CALLS	#2, DUMPSPUT_FAO_	:	
			64	0105	A2	3C	00099	MOVZWL	16(R2), -(SP)	:	0934
			7E		C3	9F	0009D	PUSHAB	P.ABX	:	
			64		02	FB	000A1	CALLS	#2, DUMPSPUT_FAO_	:	
			7E	12	A2	3C	000A4	MOVZWL	18(R2), -(SP)	:	0936
			64	0126	C3	9F	000A8	PUSHAB	P.ABY	:	
			7E		02	FB	000AC	CALLS	#2, DUMPSPUT_FAO_	:	
			64	014A	A2	3C	000AF	MOVZWL	20(R2), -(SP)	:	0938
			7E		C3	9F	000B3	PUSHAB	P.ABZ	:	
			64		02	FB	000B7	CALLS	#2, DUMPSPUT_FAO_	:	
			7E	1E	A2	3C	000BA	MOVZWL	30(R2), -(SP)	:	0940
			64	016B	C3	9F	000BE	PUSHAB	P.ACA	:	
			7E		02	FB	000C2	CALLS	#2, DUMPSPUT_FAO_	:	
			64		04	000C5	RET			:	0941

; Routine Size: 198 bytes, Routine Base: \$CODE\$ + 03A9

```

840 0942 1 GLOBAL ROUTINE DUMPS$ONE_HEADER(HEADER)=
841 0943 1
842 0944 1 !++
843 0945 1
844 0946 1 FUNCTIONAL DESCRIPTION:
845 0947 1 This routine dumps one block as a file header if it is valid.
846 0948 1
847 0949 1 INPUT PARAMETERS:
848 0950 1 HEADER - Pointer to block.
849 0951 1
850 0952 1 IMPLICIT INPUTS:
851 0953 1 NONE
852 0954 1
853 0955 1 OUTPUT PARAMETERS:
854 0956 1 NONE
855 0957 1
856 0958 1 IMPLICIT OUTPUTS:
857 0959 1 NONE
858 0960 1
859 0961 1 ROUTINE VALUE:
860 0962 1 True if the block was a valid file header and was dumped,
861 0963 1 otherwise false.
862 0964 1
863 0965 1 SIDE EFFECTS:
864 0966 1 NONE
865 0967 1
866 0968 1 --
867 0969 1
868 0970 2 BEGIN
869 0971 2 MAP
870 0972 2 LOCAL HEADER: REF BBLOCK; ! Pointer to file header
871 0973 2 LOCAL STRUCLEV, ! Structure level
872 0974 2 IDENT_AREA: REF BBLOCK, ! Pointer to ident area
873 0975 2 IDENT_LENGTH, ! Length of ident area
874 0976 2 MAP_AREA: REF BBLOCK, ! Pointer to map area
875 0977 2 MAP_LENGTH, ! Length of map area
876 0978 2 ACE_POINTER : REF BBLOCK, ! Pointer to current ACE
877 0979 2 ACE_BINDESC : BBLOCK [8], ! Descriptor to binary ACE
878 0980 2 ACE_TXTDESC : BBLOCK [8], ! Descriptor to converted ACE
879 0981 2 ACE_TEXT : BBLOCK [512]; ! Storage for converted ACE
880 0982 2 BIND
881 0983 2 FCH_CODES = PLIT(
882 0984 2 0,
883 0985 2 UPLIT BYTE (%ASCIC 'No backup'),
884 0986 2 UPLIT BYTE (%ASCIC 'Write-back cache'),
885 0987 2 UPLIT BYTE (%ASCIC 'Read check'),
886 0988 2 UPLIT BYTE (%ASCIC 'Write check'),
887 0989 2 UPLIT BYTE (%ASCIC 'Contiguous best try'),
888 0990 2 UPLIT BYTE (%ASCIC 'Deaccess locked'),
889 0991 2 UPLIT BYTE (%ASCIC 'Contiguous'),
890 0992 2 0, 0, 0,
891 0993 2 UPLIT BYTE (%ASCIC 'Corrupted Access Control List'),
892 0994 2 UPLIT BYTE (%ASCIC 'Spool file'),
893 0995 2 UPLIT BYTE (%ASCIC 'Directory'),
894 0996 2 UPLIT BYTE (%ASCIC 'Suspected bad blocks'),
895 0997 2 UPLIT BYTE (%ASCIC 'Marked for delete'),
896 0998 2

```

```

: 897      0999      2          UPLIT BYTE (%ASCIC 'Space not charged'),
: 898      1000      2          UPLIT BYTE (%ASCIC 'Erase on delete')),
: 899      1001      2          FJN_CODES = PLIT(
: 900      1002      2          UPLIT BYTE (%ASCIC 'Accessible only in recovery unit'),
: 901      1003      2          UPLIT BYTE (%ASCIC 'Enable recovery unit journal'),
: 902      1004      2          UPLIT BYTE (%ASCIC 'Before image journal'),
: 903      1005      2          UPLIT BYTE (%ASCIC 'After image journal'),
: 904      1006      2          UPLIT BYTE (%ASCIC 'Audit trail journal')),
: 905      1007      2          PLACEMENT_CODES = PLIT(
: 906      1008      2          UPLIT BYTE (%ASCIC 'Exact'),
: 907      1009      2          UPLIT BYTE (%ASCIC 'Cylinder boundary'),
: 908      1010      2          0, 0, 0, 0, 0, 0, 0, 0, 0,
: 909      1011      2          UPLIT BYTE (%ASCIC 'Specific LBN'),
: 910      1012      2          UPLIT BYTE (%ASCIC 'Specific RVN'));
: 911      1013      2          BUILTIN
: 912      1014      2          ROT;
: 913      1015      2
: 914      1016      2
: 915      1017      2          ! Validate the header.
: 916      1018      2          !
: 917      1019      2          STRUCLEV = CHECK_HEADER(.HEADER);
: 918      1020      2          IF .STRUCLEV EQL 0 THEN RETURN 0;
: 919      1021      2
: 920      1022      2          ! Initialize.
: 921      1023      2          !
: 922      1024      2          !
: 923      1025      2          LINE_DESC[0] = .DUMP$GL_WIDTH;
: 924      1026      2          LINE_DESC[1] = .DUMP$GL_OUTDESC[DESC$A POINTER];
: 925      1027      2          IDENT AREA = .HEADER + .HEADER[FH2$B IDOFFSET]*2;
: 926      1028      2          MAP AREA = .HEADER + .HEADER[FH2$B MPOFFSET]*2;
: 927      1029      2          IDENT LENGTH = .MAP AREA - .IDENT AREA;
: 928      1030      2          MAP_LENGTH = HEADER[FH2$W_CHECKSUM] - .MAP_AREA;
: 929      1031      2
: 930      1032      2
: 931      1033      2          ! If the header is valid, dump it in the proper format.
: 932      1034      2          !
: 933      1035      2          DUMP$PUT_LINE($DESCRIPTOR('Header area'));
: 934      1036      2          P 1036      2          DUMP$PUT_FAO(' Identification area offset:!11* !UL',
: 935      1037      2          .HEADER[FH2$B IDOFFSET]);
: 936      1038      2          P 1038      2          DUMP$PUT_FAO(' Map area offset:!22* !UL',
: 937      1039      2          .HEADER[FH2$B MPOFFSET]);
: 938      1040      2          IF .STRUCLEV EQL 2
: 939      1041      2          THEN
: 940      1042      2          BEGIN
: 941      1043      2          LOCAL
: 942      1044      2          P:          REF BBLOCK;
: 943      1045      2
: 944      1046      2          P 1046      2          DUMP$PUT_FAO(' Access control area offset:!11* !UL',
: 945      1047      2          .HEADER[FH2$B_ACOFFSET]);
: 946      1048      2          P 1048      2          DUMP$PUT_FAO(' Reserved area offset:!17* !UL',
: 947      1049      2          .HEADER[FH2$B_RSOFFSET]);
: 948      1050      2          P 1050      2          DUMP$PUT_FAO(' Extension segment number:!13* !UL',
: 949      1051      2          .HEADER[FH2$W_SEG_NUM]);
: 950      1052      2          P 1052      2          DUMP$PUT_FAO(' Structure level and version:!10* !UL, !UL',
: 951      1053      2          .HEADER[FH2$B_STRUCLEV],
: 952      1054      2          .HEADER[FH2$B_STRUCVER]);
: 953      1055      2          P 1055      2          DUMP$PUT_FAO(' File identification:!18* (!UL,!UL,!UL)',

```

```

: 954 P 1056 3 .HEADER[FH2$W_FID_NUM] + .HEADER[FH2$B_FID_NMX]^16,
: 955 P 1057 3 .HEADER[FH2$W_FID_SEQ],
: 956 1058 3 .HEADER[FH2$B_FID_RVN]);
: 957 P 1059 3 DUMPSPUT FAO(' Extension file identification:!8* (!UL,'UL,!UL)',
: 958 P 1060 3 .HEADER[FH2$W_EX_FIDNUM] + .HEADER[FH2$B_EX_FIDNMX]^16,
: 959 P 1061 3 .HEADER[FH2$W_EX_FIDSEQ],
: 960 1062 3 .HEADER[FH2$B_EX_FIDRVN]);
: 961 1063 3 FORMAT RMS_ATTRIBUTES(HEADER[FH2$W_RECATTR]);
: 962 1064 3 DUMPSFAO(' File characteristics:!17* ');
: 963 1065 3 FORMAT_BIT_STRING(.HEADER[FH2$L_FILECHAR], FCH_CODES);
: 964 C 1066 3 % (
: 965 C 1067 3 DUMPSFAO(' Record protection:!20* ');
: 966 C 1068 3 FORMAT_PROTECTION(.HEADER[FH2$W_RECPROT], 'RWCD');
: 967 1069 3 ) %
: 968 P 1070 3 DUMPSPUT FAO(' Map area words in use:!16* !UL',
: 969 1071 3 .HEADER[FH2$B_MAP_INUSE]);
: 970 P 1072 3 DUMPSPUT FAO(' Access mode:!26* !UL',
: 971 1073 3 .HEADER[FH2$B_ACC_MODE]);
: 972 P 1074 3 DUMPSPUT FAO(' File owner UIC:!23* !%I',
: 973 1075 3 .HEADER[FH2$L_FILEOWNER]);
: 974 1076 3 DUMPSFAO(' File protection:!22* ');
: 975 1077 3 FORMAT_PROTECTION(.HEADER[FH2$W_FILEPROT], 'RWED');
: 976 P 1078 3 DUMPSPUT FAO(' Back link file identification:!8* (!UL,'UL,!UL)',
: 977 P 1079 3 .HEADER[FH2$W_BK_FIDNUM] + .HEADER[FH2$B_BK_FIDNMX]^16,
: 978 P 1080 3 .HEADER[FH2$W_BK_FIDSEQ],
: 979 1081 3 .HEADER[FH2$B_BK_FIDRVN]);
: 980 1082 3 DUMPSFAO(' Journal control flags:!16* ');
: 981 1083 3 FORMAT_BIT_STRING(.HEADER[FH2$W_JOURNAL], FJN_CODES);
: 982 P 1084 3 DUMPSPUT FAO(' Highest block written:!16* !UL',
: 983 P 1085 3 (IF .HEADER[FH2$L_HIGHWATER] GTR 0
: 984 1086 3 THEN .HEADER[FH2$L_HIGHWATER] - 1 ELSE 0));
: 985 1087 3 DUMPSBLANK_LINE();
: 986 1088 3 DUMPSPUT_LINE($DESCRIPTOR('Identification area'));
: 987 1089 3 IF .IDENT_LENGTH GEQU $BYTEOFFSET(FI2$W_REVISION)
: 988 1090 3 THEN
: 989 1091 3 IF .HEADER[FH2$B_MPOFFSET] - .HEADER[FH2$B_IDOFFSET]
: 990 1092 3 GEQU ($BYTEOFFSET(FI2$T_FILENAMEEXT) + FI2$S_FILENAMEEXT) / 2
: 991 1093 3 THEN
: 992 P 1094 3 DUMPSPUT FAO(' File name:!28* !AF!AF',
: 993 P 1095 3 FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAME],
: 994 1096 3 FI2$S_FILENAMEEXT, IDENT_AREA[FI2$T_FILENAMEEXT])
: 995 1097 3 ELSE
: 996 P 1098 3 DUMPSPUT FAO(' File name:!28* !AF',
: 997 1099 3 FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAME]);
: 998 1100 3 IF .IDENT_LENGTH GEQU $BYTEOFFSET(FI2$Q_CREDATE)
: 999 1101 3 THEN
: 1000 P 1102 3 DUMPSPUT FAO(' Revision number:!22* !UL',
: 1001 1103 3 IDENT_AREA[FI2$W_REVISION]);
: 1002 1104 3 IF .IDENT_LENGTH GEQU $BYTEOFFSET(FI2$Q_REVDATE)
: 1003 1105 3 THEN
: 1004 1106 3 BEGIN
: 1005 1107 3 DUMPSFAO(' Creation date:!24* ');
: 1006 1108 3 FORMAT_DATE_VALUE(IDENT_AREA[FI2$Q_CREDATE]);
: 1007 1109 3 END;
: 1008 1110 3 IF .IDENT_LENGTH GEQU $BYTEOFFSET(FI2$Q_EXPDATE)
: 1009 1111 3 THEN
: 1010 1112 3 BEGIN

```

1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067

```

1113 4      DUMPSFAO(' Revision date:!24* ');
1114 4      FORMAT_DATE_VALUE(IDENT_AREA[F12$Q_REVDATE]);
1115 3      END;
1116 3      IF .IDENT_LENGTH GEQU $BYTEOFFSET(F12$Q_BAKDATE)
1117 3      THEN
1118 4      BEGIN
1119 4      DUMPSFAO(' Expiration date:!22* ');
1120 4      FORMAT_DATE_VALUE(IDENT_AREA[F12$Q_EXPDATE]);
1121 3      END;
1122 3      IF .IDENT_LENGTH GEQU F12$C_LENGTH
1123 3      THEN
1124 4      BEGIN
1125 4      DUMPSFAO(' Backup date:!26* ');
1126 4      FORMAT_DATE_VALUE(IDENT_AREA[F12$Q_BAKDATE]);
1127 3      END;
1128 3      DUMPSBLANK LINE();
1129 3      DUMSPUT_LINE($DESCRIPTOR('Map area'));
1130 3      DUMSPUT_LINE($DESCRIPTOR(' Retrieval pointers'));
1131 3      P = .MAP_AREA;
1132 3      UNTIL .P GEQA .MAP_AREA + .HEADER[FH2$B_MAP_INUSE]*2 DO
1133 4      BEGIN
1134 4      IF .P[FM2$V_FORMAT] EQL FM2$C_PLACEMENT
1135 4      THEN
1136 5      BEGIN
1137 5      DUMPSFAO('!8* Placement control: ');
1138 5      FORMAT_BIT_STRING(.P[0,0,14,0], PLACEMENT_CODES);
1139 5      P = .P + 2;
1140 5      END
1141 4      ELSE
1142 5      BEGIN
1143 5      LOCAL
1144 5      COUNT, LBN;
1145 5
1146 5      CASE .P[FM2$V_FORMAT] FROM FM2$C_FORMAT1 TO FM2$C_FORMAT3 OF
1147 5      SET
1148 5
1149 5      [FM2$C_FORMAT1]:
1150 6      BEGIN
1151 6      COUNT = .P[FM2$B_COUNT1];
1152 6      LBN = .P[FM2$W_LOWLBN];
1153 6      LBN<16,16> = .P[FM2$V_HIGHLBN];
1154 6      P = .P + 4;
1155 5      END;
1156 5
1157 5      [FM2$C_FORMAT2]:
1158 6      BEGIN
1159 6      COUNT = .P[FM2$V_COUNT2];
1160 6      LBN = .P[FM2$L_LBN2];
1161 6      P = .P + 6;
1162 5      END;
1163 5
1164 5      [FM2$C_FORMAT3]:
1165 6      BEGIN
1166 6      COUNT = ROT(.P, 16) AND (1^30-1);
1167 6      LBN = .P[FM2$L_LBN3];
1168 6      P = .P + 8;
1169 5      END;

```

```

: 1068      1170      5
: 1069      1171      5
: 1070      1172      5
: 1071      1173      5
: 1072      1174      5
: 1073      1175      5
: 1074      1176      4
: 1075      1177      3
: 1076      1178      3
: 1077      1179      3
: 1078      1180      4
: 1079      1181      4
: 1080      1182      4
: 1081      1183      4
: 1082      1184      4
: 1083      1185      4
: 1084      1186      4
: 1085      1187      4
: 1086      1188      4
: 1087      1189      5
: 1088      1190      5
: 1089      1191      5
: 1090      1192      5
: 1091      1193      5
: 1092      1194      5
: 1093      1195      5
: 1094      1196      5
: 1095      1197      5
: 1096      1198      5
: 1097      1199      5
: 1098      1200      5
: 1099      1201      5
: 1100      1202      4
: 1101      1203      3
: 1102      1204      3
: 1103      1205      3
: 1104      1206      4
: 1105      1207      4
: 1106      1208      4
: 1107      1209      4
: 1108      1210      4
: 1109      1211      5
: 1110      1212      5
: 1111      1213      5
: 1112      1214      6
: 1113      1215      6
: 1114      1216      6
: 1115      1217      6
: 1116      1218      5
: 1117      1219      6
: 1118      1220      6
: 1119      1221      6
: 1120      1222      6
: 1121      1223      5
: 1122      1224      6
: 1123      1225      6
: 1124      1226      6

```

```

TES;
DUMP$PUT FAO('!8* Count: !10UL!8* LBN: !10UL',
             .COUNT + 1,
             .LBN);
END;
END;
IF .HEADER[FH2$B_ACOFFSET] NEQ .HEADER[FH2$B_RSOFFSET]
THEN
BEGIN
DUMP$BLANK LINE ();
DUMP$PUT LINE ($DESCRIPTOR ('Access Control List'));
ACE_POINTER = .HEADER + .HEADER[FH2$B_ACOFFSET]*2;
CH$FILL (0, 8, ACE_BINDESC);
CH$FILL (0, 8, ACE_TXTDESC);
UNTIL .ACE_POINTER[ACE$B_SIZE] EQL 0
OR .ACE_POINTER GEQA .HEADER + .HEADER[FH2$B_RSOFFSET]*2
DO
BEGIN
ACE_BINDESC[DSC$W_LENGTH] = .ACE_POINTER[ACE$B_SIZE];
ACE_BINDESC[DSC$A_POINTER] = .ACE_POINTER;
ACE_TXTDESC[DSC$W_LENGTH] = 512;
ACE_TXTDESC[DSC$A_POINTER] = ACE_TEXT;
$FORMAT_ACL (ACLENT = ACE_BINDESC,
             ACLLEN = ACE_TXTDESC[DSC$W_LENGTH],
             ACLSTR = ACE_TXTDESC,
             WIDTH = %REF (80),
             TRMDSC = $DESCRIPTOR (%CHAR (13), %CHAR (10)),
             INDENT = %REF (4));
DUMP$PUT LINE (ACE_TXTDESC);
ACE_POINTER = .ACE_POINTER + .ACE_POINTER[ACE$B_SIZE];
END;
IF .HEADER[FH2$B_RSOFFSET] NEQ $BYTEOFFSET (FH2$W_CHECKSUM) / 2
THEN
BEGIN
DUMP$BLANK LINE ();
DUMP$PUT LINE ($DESCRIPTOR ('User reserved area'));
P = .HEADER + .HEADER[FH2$B_RSOFFSET]*2;
DO
BEGIN
CASE HEADER[FH2$W_CHECKSUM] - .P FROM 1 TO 3 OF
SET
[1]:
BEGIN
DUMP$PUT_FAO ('!XB '!AF'',
             .P[0, 0, 8, 0], 1, .P);
P = .P + 1;
END;
[2]:
BEGIN
DUMP$PUT_FAO ('!XW '!AF'',
             .P[0, 0, 16, 0], 2, .P);
P = .P + 2;
END;
[3]:
BEGIN
DUMP$PUT_FAO ('!6XL '!AF'',
             .P[0, 0, 24, 0], 3, .P);

```

: 1125
: 1126
: 1127
: 1128
: 1129
: 1130
: 1131
: 1132
: 1133
: 1134
: 1135
: 1136
: 1137
: 1138
: 1139
: 1140
: 1141
: 1142
: 1143
: 1144
: 1145
: 1146
: 1147
: 1148
: 1149
: 1150
: 1151
: 1152
: 1153
: 1154
: 1155
: 1156
: 1157
: 1158
: 1159
: 1160
: 1161
: 1162
: 1163
: 1164
: 1165
: 1166
: 1167
: 1168
: 1169
: 1170
: 1171
: 1172
: 1173
: 1174
: 1175
: 1176
: 1177
: 1178
: 1179
: 1180
: 1181

```

        P = .P + 3;
        END;
        [OUTRANGE]:
        BEGIN
        DUMPSPUT_FAO ('      !XL  ''!AF''',
                    .P[0, 0, 32, 0], '4, .P);
        P = .P + 4;
        END;
        TES
        END
    UNTIL .P GEQA HEADER[FH2$W_CHECKSUM];
    END;
ELSE
    BEGIN
    LOCAL
    P:
        REF BBLOCK,
        FILE_NAME:
            VECTOR[20, BYTE],
        FILE_LENGTH;

    DUMPSPUT_FAO('      File identification:!18* (!UL,!UL)',
                .HEADER[FH1$W_FID_NUM],
                .HEADER[FH1$W_FID_SEQ]);
    DUMPSPUT_FAO('      Structure level and version:!10* !UL, !UL',
                .HEADER[FH2$B_STRUCLEV],
                .HEADER[FH2$B_STRUCVER]);
    DUMPSPUT_FAO('      File owner UIC:!23* !%I',
                .HEADER[FH1$B_UICGROUP]^16 OR .HEADER[FH1$B_UICMEMBER]);
    DUMPSFAO('      File protection:!22* ');
    FORMAT_PROTECTION(.HEADER[FH1$W_FILEPROT], 'RWED');
    DUMPSFAO('      File characteristics:!17* ');
    FORMAT_BIT_STRING(.HEADER[FH1$W_FILECHAR], FCH_CODES);
    FORMAT_RMS_ATTRIBUTES(HEADER[FH1$W_RECATTR]);
    DUMPSBLANK_LINE();
    DUMPSPUT_LINE($DESCRIPTOR('Identification area'));
    FILE_LENGTH = MAKE_STRING(
        IDENT_AREA[F11$W_FILENAME] - $BYTEOFFSET(NMBS$W_NAME),
        FILE_NAME);
    DUMPSPUT_FAO('      File name:!28* !AF',
                .FILE_LENGTH, FILE_NAME);
    DUMPSPUT_FAO('      Revision number:!22* !UL',
                .IDENT_AREA[F11$W_REVISION]);
    DUMPSFAO('      Revision date:!24* ');
    FORMAT_ODS1_DATE_VALUE(IDENT_AREA[F11$T_REVDATE]);
    DUMPSFAO('      Creation date:!24* ');
    FORMAT_ODS1_DATE_VALUE(IDENT_AREA[F11$T_CREDATE]);
    DUMPSFAO('      Expiration date:!22* ');
    FORMAT_ODS1_DATE_VALUE(IDENT_AREA[F11$T_EXPDATE]);
    DUMPSBLANK_LINE();
    DUMPSPUT_LINE($DESCRIPTOR('Map area'));
    DUMPSPUT_FAO('      Extension segment number:!13* !UL',
                .MAP_AREA[FM1$B_EX_SEGNUM]);
    DUMPSPUT_FAO('      Extension file identification:!8* (!UL,!UL,!UL)',
                .MAP_AREA[FM1$W_EX_FILNUM],
                .MAP_AREA[FM1$W_EX_FILSEQ],
                .MAP_AREA[FM1$B_EX_RVN]);
    DUMPSPUT_FAO('      Retrieval pointer count size:!9* !UL',

```

```

: 1182      1284      3
: 1183      P 1285      3 DUMP$PUT_FAO(' Retrieval pointer LBN size:!11* !UL',
: 1184      1286      3 .MAP_AREA[FM1$B_LBNSIZE]);
: 1185      P 1287      3 DUMP$PUT_FAO(' Map area words in use:!16* !UL',
: 1186      1288      3 .MAP_AREA[FM1$B_INUSE]);
: 1187      P 1289      3 DUMP$PUT_FAO(' Map area words available:!13* !UL',
: 1188      1290      3 .MAP_AREA[FM1$B_AVAIL]);
: 1189      1291      3 DUMP$PUT_LINE($DESCRIPTOR(' Retrieval pointers'));
: 1190      1292      3 P = .MAP_AREA + FM1$C_POINTERS;
: 1191      1293      3 UNTIL .P GEQA .MAP_AREA + FM1$C_POINTERS + .MAP_AREA[FM1$B_INUSE]*2 DO
: 1192      1294      3 BEGIN
: 1193      P 1295      3 DUMP$PUT_FAO('!8* Count: !10UL!8* LBN: !10UL',
: 1194      P 1296      3 .P[FM1$B_COUNT] + 1,
: 1195      1297      3 .P[FM1$W_LOWLBN] + .P[FM1$B_HIGHLBN]^16);
: 1196      1298      3 P = .P + 4;
: 1197      1299      3 END;
: 1198      1300      3 END;
: 1199      1301      3 DUMP$BLANK LINE();
: 1200      P 1302      3 DUMP$PUT_FAO('Checksum:!33* !UL',
: 1201      1303      3 .HEADER[FH2$W_CHECKSUM]);
: 1202      1304      3
: 1203      1305      3
: 1204      1306      2 RETURN 1;
: 1205      1307      1 END;

```

														.PSECT		\$SPLITS, NOWRT, NOEXE, 2		
63	61	63	20	68	63	61	62	2D	65	74	69	72	57	10	00380	P.ACC:	.ASCII	<9>\No backup\
													65	68	0038A	P.ACD:	.ASCII	<16>\Write-back cache\
				6B	63	65	68	63	20	64	61	65	52	0A	00399			
73	65	62	20	73	75	6F	75	67	69	74	6E	6F	43	13	0039B	P.ACE:	.ASCII	<10>\Read check\
													0B	00	003A6	P.ACF:	.ASCII	<11>\Write check\
													43	13	003B2	P.ACG:	.ASCII	<19>\Contiguous best try\
65	6B	63	6F	6C	20	73	73	65	63	63	61	65	44	0F	003C1			
													64	0F	003C6	P.ACH:	.ASCII	<15>\Deaccess locked\
				73	75	6F	75	67	69	74	6E	6F	43	0A	003D5			
65	63	63	41	20	64	65	74	70	75	72	72	6F	43	1D	003D6	P.ACI:	.ASCII	<10>\Contiguous\
74	73	69	4C	20	6C	6F	72	74	6E	6F	43	20	73	73	003E1	P.ACJ:	.ASCII	<29>\Corrupted Access Control List\
					65	6C	69	66	20	6C	6F	6F	70	53	003F0			
					79	72	6F	74	63	65	72	69	44	09	003FF	P.ACK:	.ASCII	<10>\Spool file\
20	64	61	62	20	64	65	74	63	65	70	73	75	53	14	0040A	P.ACL:	.ASCII	<9>\Directory\
													61	6C	00414	P.ACM:	.ASCII	<20>\Suspected bad blocks\
6C	65	64	20	72	6F	66	20	64	65	6B	72	61	4D	11	00423			
													65	74	00429	P.ACN:	.ASCII	<17>\Marked for delete\
72	61	68	63	20	74	6F	6E	20	65	63	61	70	53	11	00438			
													64	65	0043B	P.ACO:	.ASCII	<17>\Space not charged\
74	65	6C	65	64	20	6E	6F	20	65	73	61	72	45	0F	0044A			
													65	0F	0044D	P.ACP:	.ASCII	<15>\Erase on delete\
													65	0F	0045C			
															0045D			
															00460		.BLKB	3
															00464	P.ACB:	.LONG	18
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00468		.LONG	0
															00480		.ADDRESS	P.ACC, P.ACD, P.ACE, P.ACF, P.ACG, -
															00484		P.ACH, P.ACI	
																	.LONG	0, 0, 0


```

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00490 .ADDRESS P.ACJ, P.ACK, P.ACL, P.ACM, P.ACN, -
00000000' 00000000' 00000000' 00000000' 00000000' 004A8 P.ACO, P.ACP
6C 6E 6F 20 65 6C 62 69 73 73 65 63 63 41 20 004AC P.ACR: .ASCII \ Accessible only in recovery unit\
75 20 79 72 65 76 6F 63 65 72 20 6E 69 20 79 004BB
72 65 76 6F 63 65 72 20 65 6C 62 61 6E 45 1C 004CA
6C 61 6E 72 75 6F 6A 20 74 69 6E 75 20 79 004CD P.ACS: .ASCII <28>\Enable recovery unit journal\
6A 20 65 67 61 6D 69 20 65 72 6F 66 65 42 14 004DC P.ACT: .ASCII <20>\Before image journal\
6F 6A 20 65 67 61 6D 69 20 72 6C 61 6E 72 75 6F 004EA P.ACU: .ASCII <19>\After image journal\
6F 6A 20 6C 69 61 72 74 20 74 6C 61 6E 72 75 004FF P.ACV: .ASCII <19>\Audit trail journal\
0050E
00513
00522
00527 .BLKB 1
00528 .LONG 5
00000005
00000000' 00000000' 00000000' 00000000' 00000000' 0052C P.ACQ: .ADDRESS P.ACR, P.ACS, P.ACT, P.ACU, P.ACV
64 6E 75 6F 62 20 72 65 64 6E 63 61 78 45 05 00540 P.ACX: .ASCII <5>\Exact\
4E 42 4C 20 63 69 66 69 63 65 70 53 0C 00546 P.ACY: .ASCII <17>\Cylinder boundary\
4E 56 52 20 63 69 66 69 63 65 70 53 0C 00555
00558 P.ACZ: .ASCII <12>\Specific LBN\
00565 P.ADA: .ASCII <12>\Specific RVN\
00572 .BLKB 2
00574 .LONG 14
0000000E
00000000' 00000000' 00000000' 00000000' 00000000' 00578 P.ACW: .ADDRESS P.ACX, P.ACY
00000000' 00000000' 00000000' 00000000' 00000000' 00580 .LONG 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
00000000' 00000000' 00000000' 00000000' 00000000' 00598
00000000' 00000000' 00000000' 00000000' 00000000' 005A8 .ADDRESS P.ACZ, P.ADA
61 65 72 61 20 72 65 64 61 65 48 005B0 P.ADC: .ASCII \Header area\
005BB .BLKB 1
0000000B
00000000' 00000000' 00000000' 00000000' 00000000' 005BC P.ADB: .LONG 11
005C0 .ADDRESS P.ADC
61 63 69 66 69 74 6E 65 64 49 20 20 20 20 27 005C4 P.ADD: .ASCII \ Identification area offset:!1* !UL\
65 73 66 66 6F 20 61 65 72 61 20 6E 6F 69 74 005D3
4C 55 21 20 2A 31 31 21 3A 74 005E2
6F 20 61 65 72 61 20 70 61 4D 20 20 20 20 1C 005EC P.ADE: .ASCII <28>\ Map area offset:!22* !UL\
6E 6F 63 20 73 73 65 63 63 41 20 20 20 20 27 00609 P.ADF: .ASCII \ Access control area offset:!1* !UL\
65 73 66 66 6F 20 61 65 72 61 20 6C 6F 72 74 00618
4C 55 21 20 2A 31 31 21 3A 74 00627
61 20 64 65 76 72 65 73 65 52 20 20 20 20 21 00631 P.ADG: .ASCII \ Reserved area offset:!17* !UL\
2A 37 31 21 3A 74 65 73 66 66 6F 20 61 65 72 00640
4C 55 21 20 0064F
20 6E 6F 69 73 6E 65 74 78 45 20 20 20 20 25 00653 P.ADH: .ASCII \% Extension segment number:!13* !UL\
3A 72 65 62 6D 75 6E 20 74 6E 65 6D 67 65 73 00662
4C 55 21 20 2A 33 31 21 00671
20 65 72 75 74 63 75 72 74 53 20 20 20 20 2D 00679 P.ADI: .ASCII \- Structure level and version:!10* !UL\
69 73 72 65 76 20 64 6E 61 20 6C 65 76 65 6C 00688
55 21 2U 2A 30 31 21 3A 6E 6F 00697
4C 55 21 20 2C 4C 006A1 .ASCII \L, !UL\
74 6E 65 64 69 20 65 6C 69 46 20 20 20 20 2A 006A7 P.ADJ: .ASCII \* File identification:!18* (!UL,!UL,!UL\
20 2A 38 31 21 3A 6E 6F 69 74 61 63 69 66 69 006B6
21 2C 4C 55 21 2C 4C 55 21 2C 4C 55 21 28 006C5
29 4C 55 006CF
20 6E 6F 69 73 6E 65 74 78 45 20 20 20 20 33 006D2 P.ADK: .ASCII \UL)\
61 63 69 66 69 74 6E 65 64 69 20 65 6C 69 66 006E1 .ASCII \3 Extension file identification:!8* (\
28 20 2A 38 21 3A 6E 6F 69 74 006F0
29 4C 55 21 2C 4C 55 21 2C 4C 55 21 006FA .ASCII \!UL,!UL,!UL)\

```

61	72	61	68	63	20	65	6C	69	46	20	20	20	20	1E	00706	P.ADL:	.ASCII	<30>\	File characteristics:!17* \	
2A	37	31	21	3A	73	63	69	74	73	69	72	65	74	63	00715					
77	20	61	65	72	61	20	70	61	4D	20	20	20	20	22	00724					
36	31	21	3A	65	73	75	20	6E	69	20	73	64	72	6F	00725	P.ADM:	.ASCII	'	Map area words in use:!16* !UL\	
64	6F	6D	20	73	73	65	63	63	41	20	20	20	20	18	00734					
72	65	6E	77	6F	20	65	6C	69	46	20	20	20	20	1B	00743	P.ADN:	.ASCII	<24>\	Access mode:!26* !UL\	
65	74	6F	72	70	20	65	6C	69	46	20	20	20	20	19	00757					
20	6B	6E	69	6C	20	6B	63	61	42	20	20	20	20	33	00761	P.ADO:	.ASCII	<27>\	File owner UIC:!23* !%I\	
61	63	69	66	69	74	6E	65	64	69	20	20	65	6C	69	66	00770	P.ADP:	.ASCII	<25>\	File protection:!22* \
6F	63	20	6C	61	6E	72	75	6F	4A	20	2C	20	20	1F	0077D					
36	31	21	3A	73	67	61	6C	66	20	6C	6F	72	74	6E	0078C	P.ADQ:	.ASCII	\3	Back link file identification:!8* (\	
6C	62	20	74	73	65	68	67	69	48	20	20	20	20	22	007A6					
36	31	21	3A	6E	65	74	74	69	72	77	20	6B	63	6F	007B5	P.ADR:	.ASCII	\!UL,!UL,!UL)\	Journal control flags:!16* \	
20	6E	6F	69	74	61	63	69	66	69	74	6E	65	64	49	007CB	P.ADR:	.ASCII	<31>\	Highest block written:!16* !UL\	
3A	65	6D	61	6E	20	65	6C	69	46	20	20	20	20	19	007DA	P.ADU:	.ASCII	\	Identification area\	
3A	65	6D	61	6E	20	65	6C	69	46	20	20	20	20	19	007E9					
6E	20	6E	6F	69	73	69	76	65	52	20	20	20	20	1C	007EB	P.ADT:	.BLKB	3		
64	20	6E	6F	69	74	61	65	72	43	20	20	20	20	17	007FA	P.ADT:	.LONG	19		
64	20	6E	6F	69	73	69	76	65	52	20	20	20	20	17	0080E	P.ADT:	.ADDRESS	P.ADU		
6E	6F	69	74	61	72	69	70	78	45	20	20	20	20	19	00821	P.ADV:	.ASCII	<25>\	File name:!28* !AF!AF\	
74	61	64	20	70	75	6B	63	61	42	20	20	20	20	15	00828	P.ADV:	.ASCII	<22>\	File name:!28* !AF\	
20	74	6E	65	6D	65	63	61	6C	50	20	2A	38	21	1B	00855	P.ADX:	.ASCII	<28>\	Revision number:!22* !UL\	
30	31	21	20	3A	74	6E	75	6F	43	20	2A	38	21	1E	0087A	P.ADY:	.ASCII	<23>\	Creation date:!24* \	
55	30	31	21	20	3A	4E	42	4C	20	2A	38	21	4C	55	00889	P.ADZ:	.ASCII	<23>\	Revision date:!24* \	
20	6C	6F	72	74	6E	6F	43	20	73	73	65	63	63	41	008A1	P.AEA:	.ASCII	<25>\	Expiration date:!22* \	
															008AA	P.AEB:	.ASCII	<21>\	Backup date:!26* \	
															008B9					
															008C4	P.AED:	.ASCII	\Map area\		
															008D3					
															008DA	P.AEC:	.BLKB	2		
															008E2	P.AEC:	.LONG	8		
															008E4	P.AEF:	.ADDRESS	P.AED		
															008E8	P.AEF:	.ASCII	\	Retrieval pointers\	
															008EB					
															00902	P.AEE:	.BLKB	2		
															00904	P.AEE:	.LONG	22		
															00908	P.AEE:	.ADDRESS	P.AEF		
															0090C	P.AEG:	.ASCII	<24>\!8*	Placement control: \	
															0091B					
															00925	P.AEH:	.ASCII	<30>\!8*	Count: !10UL!8* LBN: !10UL\	
															00934					
															00943					
															00944	P.AEJ:	.ASCII	\Access Control List\		
															00953					

	6A		02	FB	00069	CALLS	#2, DUMP\$PUT_FAO_	1039
			54	DD	0006C	PUSHL	R4	
		0188	CB	9F	0006E	PUSHAB	P.ADE	
	6A		02	FB	00072	CALLS	#2, DUMP\$PUT_FAO_	
	54		A7	9E	00075	MOVAB	14(R7), R4	1062
	02		0E	58	D1 00079	CMPL	STRUCLÉV, #2	1040
			03	13	0007C	BEQL	2\$	
			0357	31	0007E	BRW	29\$	
	7E		02	A7	9A 00081	MOVZBL	2(R7), -(SP)	1047
			01A5	CB	9F 00085	PUSHAB	P.ADF	
	6A		02	FB	00089	CALLS	#2, DUMP\$PUT_FAO_	
	7E		03	A7	9A 0008C	MOVZBL	3(R7), -(SP)	1049
			01CD	CB	9F 00090	PUSHAB	P.ADG	
	6A		02	FB	00094	CALLS	#2, DUMP\$PUT_FAO_	
	7E		04	A7	3C 00097	MOVZWL	4(R7), -(SP)	1051
			01EF	CB	9F 0009B	PUSHAB	P.ADH	
	6A		02	FB	0009F	CALLS	#2, DUMP\$PUT_FAO_	
	7E		06	A7	9A 000A2	MOVZBL	6(R7), -(SP)	1054
	7E		07	A7	9A 000A6	MOVZBL	7(R7), -(SP)	
			0215	CB	9F 000AA	PUSHAB	P.ADI	
	6A		03	FB	000AE	CALLS	#3, DUMP\$PUT_FAO_	
	7E		0C	A7	9A 000B1	MOVZBL	12(R7), -(SP)	1058
	7E		0A	A7	3C 000B5	MOVZWL	10(R7), -(SP)	
	50		08	A7	3C 000B9	MOVZWL	8(R7), R0	
	51		0D	A7	9A 000BD	MOVZBL	13(R7), R1	
51	51		10	78	000C1	ASHL	#16, R1, R1	
			6140	9F	000C5	PUSHAB	(R1)[R0]	
			0243	CB	9F 000C8	PUSHAB	P.ADJ	
	6A		04	FB	000CC	CALLS	#4, DUMP\$PUT_FAO_	
	7E		12	A7	9A 000CF	MOVZBL	18(R7), -(SP)	1062
	7E		10	A7	3C 000D3	MOVZWL	16(R7), -(SP)	
	50		64	3C	000D7	MOVZWL	(R4), R0	
	51		13	A7	9A 000DA	MOVZBL	19(R7), R1	
51	51		10	78	000DE	ASHL	#16, R1, R1	
			6140	9F	000E2	PUSHAB	(R1)[R0]	
			026E	CB	9F 000E5	PUSHAB	P.ADK	
	6A		04	FB	000E9	CALLS	#4, DUMP\$PUT_FAO_	
			14	A7	9F 000EC	PUSHAB	20(R7)	1063
0175	CA		01	FB	000EF	CALLS	#1, FORMAT_RMS_ATTRIBUTES	1064
			02A2	CB	9F 000F4	PUSHAB	P.ADL	
A0	AA		01	FB	000F8	CALLS	#1, DUMP\$FAO_	1065
			5B	DD	000FC	PUSHL	R11	
			34	A7	DD 000FE	PUSHL	52(R7)	
0B	AA		02	FB	00101	CALLS	#2, FORMAT_BIT_STRING	1071
	54		3A	A7	9A 00105	MOVZBL	58(R7), R4	
			54	DD	00109	PUSHL	R4	
			02C1	CB	9F 0010B	PUSHAB	P.ADM	
6A			02	FB	0010F	CALLS	#2, DUMP\$PUT_FAO_	
7E			3B	A7	9A 00112	MOVZBL	59(R7), -(SP)	1073
			02E4	CB	9F 00116	PUSHAB	P.ADN	
6A			02	FB	0011A	CALLS	#2, DUMP\$PUT_FAO_	
			3C	A7	DD 0011D	PUSHL	60(R7)	1075
			02FD	CB	9F 00120	PUSHAB	P.ADO	
6A			02	FB	00124	CALLS	#2, DUMP\$PUT_FAO_	
			0319	CB	9F 00127	PUSHAB	P.ADP	1076
A0	AA		01	FB	0012B	CALLS	#1, DUMP\$FAO_	
			44455752	8F	DD 0012F	PUSHL	#1145395026 -	1077

	0138	7E	40	A7	3C	00135	MOVZWL	64(R7), -(SP)	
		CA		02	FB	00139	CALLS	#2, FORMAT PROTECTION	1081
		7E	46	A7	9A	0013E	MOVZBL	70(R7), -(SP)	
		7E	44	A7	3C	00142	MOVZWL	68(R7), -(SP)	
		50	42	A7	3C	00146	MOVZWL	66(R7), R0	
51		51	47	A7	9A	0014A	MOVZBL	71(R7), R1	
		51		10	78	0014E	ASHL	#16, R1, R1	
				6140	9F	00152	PUSHAB	(R1)[R0]	
			0333	CB	9F	00155	PUSHAB	P.ADQ	
		6A		04	FB	00159	CALLS	#4, DUMPSPUT_FAO_	
			0367	CB	9F	0015C	PUSHAB	P.ADR	1082
	A0	AA		01	FB	00160	CALLS	#1, DUMPSFAO_	
			00C8	CB	9F	00164	PUSHAB	FJN CODES	1083
		7E	48	A7	3C	00168	MOVZWL	72(R7), -(SP)	
	OB	AA		02	FB	0016C	CALLS	#2, FORMAT_BII_STRING	
			4C	A7	DS	00170	TSTL	76(R7)	1086
				09	15	00173	BLEQ	33	
50		4C		01	C3	00175	SUBL3	#1, 76(R7), R0	
				50	DD	0017A	PUSHL	R0	
				02	11	0017C	BRB	4\$	
				7E	D4	0017E	3\$: CLRL	-(SP)	
			0387	CB	9F	00180	4\$: PUSHAB	P.ADS	
		6A		02	FB	00184	CALLS	#2, DUMSPUT_FAO	
	00000000G	00		00	FB	00187	CALLS	#0, DUMPSBLANK_LINE	1087
			03C0	CB	9F	0018E	PUSHAB	P.ADT	1088
	00000000G	00		01	FB	00192	CALLS	#1, DUMSPUT_LINE	
		14		56	D1	00199	CMPL	IDENT_LENGTH, #20	1089
				2E	1F	0019C	BLSSU	6\$	
		50	01	A7	9A	0019E	MOVZBL	1(R7), R0	1091
		51		67	9A	001A2	MOVZBL	(R7), R1	
		50		51	C2	001A5	SUBL2	R1, R0	
		3C		50	D1	001AB	CMPL	R0, #60	1092
				14	1F	001AB	BLSSU	5\$	
			36	A2	9F	001AD	PUSHAB	54(IDENT_AREA)	1096
		7E	42	8F	9A	001B0	MOVZBL	#66, -(SP)	
				52	DD	001B4	PUSHL	IDENT_AREA	
				14	DD	001B6	PUSHL	#20	
			03C8	CB	9F	001B8	PUSHAB	P.ADV	
		6A		05	FB	001BC	CALLS	#5, DUMSPUT_FAO_	
				0B	11	001BF	BRB	6\$	
				52	DD	001C1	5\$: PUSHL	IDENT_AREA	1099
				14	DD	001C3	PUSHL	#20	
			03E2	CB	9F	001C5	PUSHAB	P.ADW	
		6A		03	FB	001C9	CALLS	#3, DUMSPUT_FAO	
		16		56	D1	001CC	6\$: CMPL	IDENT_LENGTH, #22	1100
				0B	1F	001CF	BLSSU	7\$	
		7E	14	A2	3C	001D1	MOVZWL	20(IDENT_AREA), -(SP)	1103
			03F9	CB	9F	001D5	PUSHAB	P.ADX	
		6A		02	FB	001D9	CALLS	#2, DUMSPUT_FAO	
		1E		56	D1	001DC	7\$: CMPL	IDENT_LENGTH, #30	1104
				10	1F	001DF	BLSSU	8\$	
			0416	CB	9F	001E1	PUSHAB	P.ADY	1107
	A0	AA		01	FB	001E5	CALLS	#1, DUMPSFAO	
			16	A2	9F	001E9	PUSHAB	22(IDENT_AREA)	1108
	00C5	CA		01	FB	001EC	CALLS	#1, FORMAT DATE VALUE	
		26		56	D1	001F1	8\$: CMPL	IDENT_LENGTH, #38	1110
				10	1F	001F4	BLSSU	9\$	

			042E	CB	9F	001F6		PUSHAB	P.ADZ	1113
	A0	AA		01	FB	001FA		CALLS	#1, DUMPSFAO	
			1E	A2	9F	001FE		PUSHAB	30(IDENT AREA)	1114
	00C5	CA		01	FB	00201		CALLS	#1, FORMAT DATE VALUE	
		2E		56	D1	00206	9\$:	CMPL	IDENT_LENGTH, #26	1116
				10	1F	00209		BLSSU	10\$	
			0446	CB	9F	0020B		PUSHAB	P.AEA	1119
	A0	AA		01	FB	0020F		CALLS	#1, DUMPSFAO	
			26	A2	9F	00213		PUSHAB	38(IDENT AREA)	1120
	00C5	CA		01	FB	00216		CALLS	#1, FORMAT DATE VALUE	
	00000078	8F		56	D1	0021B	10\$:	CMPL	IDENT_LENGTH, #T20	1122
				10	1F	00222		BLSSU	11\$	
			0460	CB	9F	00224		PUSHAB	P.AEB	1125
	A0	AA		01	FB	00228		CALLS	#1, DUMPSFAO	
			2E	A2	9F	0022C		PUSHAB	46(IDENT AREA)	1126
	00C5	CA		01	FB	0022F		CALLS	#1, FORMAT DATE VALUE	
	00000000G	00		00	FB	00234	11\$:	CALLS	#0, DUMPSBANK_LINE	1128
			0480	CB	9F	0023B		PUSHAB	P.AEC	1129
	00000000G	00		01	FB	0023F		CALLS	#1, DUMPSPUT_LINE	
			04A0	CB	9F	00246		PUSHAB	P.AEE	1130
	00000000G	00		01	FB	0024A		CALLS	#1, DUMPSPUT_LINE	
		56		53	D0	00251		MOVL	MAP AREA, P	1131
		50		6344	3E	00254	12\$:	MOVAW	(MAP AREA)[R4], R0	1132
		50		56	D1	00258		CMPL	P, R0	
				6E	1E	0025B		BGEQU	19\$	
	CO	8F	01	A6	93	0025D		BITB	1(P), #192	1134
				1A	12	00262		BNEQ	13\$	
			04A8	CB	9F	00264		PUSHAB	P.AEG	1137
	A0	AA		01	FB	00268		CALLS	#1, DUMPSFAO	
			0114	CB	9F	0026C		PUSHAB	PLACEMENT CODES	1138
7E	66	0E		00	EF	00270		EXTZV	#0, #14, 7P, -(SP)	
		AA		02	FB	00275		CALLS	#2, FORMAT_BIT_STRING	
		56		02	CO	00279		ADDL2	#2, P	1139
				D6	11	0027C		BRB	12\$	1134
		02		0E	EF	0027E	13\$:	EXTZV	#14, #2, (P), R2	1146
52	66	01		52	CF	00283		CASEL	R2, #1, #2	
	002A	001C		0006		00287	14\$:	.WORD	15\$-14\$,- 16\$-14\$,- 17\$-14\$	
				86	9A	0028D	15\$:	MOVZBL	(P)+, COUNT	1151
		50		01	A6	3C	00290	MOVZWL	1(P), LBN	1152
55	86	06		00	EF	00294		EXTZV	#0, #6, (P)+, R5	1153
51	10	10		55	F0	00299		INSV	R5, #16, #16, LBN	
		56		02	CO	0029E		ADDL2	#2, P	1154
				1A	11	002A1		BRB	18\$	1146
50	86	0E		00	EF	002A3	16\$:	EXTZV	#0, #14, (P)+, COUNT	1159
		51		01	A6	002A8		MOVL	1(P), LBN	1160
		56		05	CO	002AC		ADDL2	#5, P	1161
				0C	11	002AF		BRB	18\$	1146
		86		10	9C	002B1	17\$:	ROTL	#16, (P)+, R1	1166
50	51	1E		00	EF	002B5		EXTZV	#0, #30, R1, COUNT	
	51	51		86	D0	002BA		MOVL	(P)+, LBN	1167
				51	DD	002BD	18\$:	PUSHL	LBN	1175
			01	A0	9F	002BF		PUSHAB	1(COUNT)	
			04C1	CB	9F	002C2		PUSHAB	P.AEH	
		6A		03	FB	002C6		CALLS	#3, DUMPSPUT_FAO_	
				89	11	002C9		BRB	12\$	1132

52	02	A7	9A	002CB	19\$:	MOVZBL	2(R7), R2	1178
58	03	A7	9A	002CF		MOVZBL	3(R7), R8	
58		52	D1	002D3		CMPL	R2, R8	
		7B	13	002D6		BEQL	21\$	
		00	FB	002D8		CALLS	#0, DUMPSBLANK_LINE	1181
00000000G	00	04F4	CB	9F 002DF		PUSHAB	P.AEI	1182
00000000G	00		01	FB 002E3		CALLS	#1, DUMSPUT_LINE	
	59	6742	3E	002EA		MOVAW	(R7)[R2], ACE_POINTER	1183
08	00	6E	00	2C 002EE		MOVCS	#0, (SP), #0, #8, ACE_BINDESC	1184
			F8	AD 002F3				
08	00	6E	00	2C 002F5		MOVCS	#0, (SP), #0, #8, ACE_TXTDESC	1185
			F0	AD 002FA				
			69	95 002FC	20\$:	TSTB	(ACE_POINTER)	1186
			53	13 002FE		BEQL	21\$	
		50	6748	3E 00300		MOVAW	(R7)[R8], R0	1187
		50	59	D1 00304		CMPL	ACE_POINTER, R0	
			4A	1E 00307		BGEQU	21\$	
	F8	AD	69	9B 00309		MOVZBW	(ACE_POINTER), ACE_BINDESC	1190
	FC	AD	59	D0 0030D		MOVL	ACE_POINTER, ACE_BINDESC+4	1191
	F0	AD	0200	8F 80 00311		MOVW	#512, ACE_TXTDESC	1192
	F4	AD	1C	AE 9E 00317		MOVAB	ACE_TEXT, ACE_TXTDESC+4	1193
			7E	D4 0031C		CLRL	-(SP)	1199
	08	AE	04	D0 0031E		MOVL	#4, 8(SP)	
		08	AE	9F 00322		PUSHAB	8(SP)	
		0500	CB	9F 00325		PUSHAB	P.AEK	
		50	8F	9A 00329		MOVZBL	#80, 12(SP)	
	0C	AE	0C	AE 9F 0032E		PUSHAB	12(SP)	
			F0	AD 9F 00331		PUSHAB	ACE_TXTDESC	
			F0	AD 9F 00334		PUSHAB	ACE_TXTDESC	
			F8	AD 9F 00337		PUSHAB	ACE_BINDESC	
00000000G	00		07	FB 0033A		CALLS	#7, SYSSFORMAT_ACL	
00000000G	00		F0	AD 9F 00341		PUSHAB	ACE_TXTDESC	1200
			01	FB 00344		CALLS	#1, DUMSPUT_LINE	
			69	9A 0034B		MOVZBL	(ACE_POINTER), R0	1201
			50	C0 0034E		ADDL2	R0, ACE_POINTER	
			A9	11 00351		BRB	20\$	1186
	FF	8F	58	91 00353	21\$:	CMPB	R8, #255	1204
			7C	13 00357		BEQL	28\$	
00000000G	00		00	FB 00359		CALLS	#0, DUMPSBLANK_LINE	1207
			051C	CB 9F 00360		PUSHAB	P.AEM	1208
00000000G	00		01	FB 00364		CALLS	#1, DUMSPUT_LINE	
			6748	3E 0036B		MOVAW	(R7)[R8], P	1209
	50	57	56	C3 0036F	22\$:	SUBL3	P, R7, R0	1212
02 FFFFE03	8F		50	CF 00373		CASEL	R0, #-509, #2	
003D 002A			0018	0037B	23\$:	.WORD	24\$-23\$, -	
							25\$-23\$, -	
							26\$-23\$	
			56	DD 00381		PUSHL	P	1231
			04	DD 00383		PUSHL	#4	
			66	DD 00385		PUSHL	(P)	
			0564	CB 9F 00387		PUSHAB	P.AER	
	6A		04	FB 0038B		CALLS	#4, DUMSPUT_FAO_	
	56		04	C0 0038E		ADDL2	#4, P	1232
			38	11 00391		BRB	27\$	1211
			56	DD 00393	24\$:	PUSHL	P	1216
			01	DD 00395		PUSHL	#1	
			7E	66 9A 00397		MOVZBL	(P), -(SP)	

		6A	0524	CB 9F 0039A	PUSHAB	P.AEO		
				04 FB 0039E	CALLS	#4, DUMP\$PUT_FAO_		1217
				56 D6 003A1	INCL	P		1211
				26 11 003A3	BRB	27\$		1221
				56 DD 003A5	25\$: PUSHL	P		
				02 DD 003A7	PUSHL	#2		
		7E		66 3C 003A9	MOVZWL	(P), -(SP)		
			053C	CB 9F 003AC	PUSHAB	P.AEP		
		6A		04 FB 003B0	CALLS	#4, DUMP\$PUT_FAO_		1222
		56		02 C0 003B3	ADDL2	#2, P		1211
				13 11 003B6	BRB	27\$		1226
				56 DD 003B8	26\$: PUSHL	P		
				03 DD 003BA	PUSHL	#3		
7E	66		18	00 EF 003BC	EXTZV	#0, #24, (P), -(SP)		
			0551	CB 9F 003C1	PUSHAB	P.AEQ		
		6A		04 FB 003C5	CALLS	#4, DUMP\$PUT_FAO_		1227
		56		03 C0 003C8	ADDL2	#3, P		1236
		50	01FE	C7 9E 003CB	27\$: MOVAB	510(R7), R0		
		50		56 D1 003D0	CMPL	P, R0		
				9A 1F 003D3	BLSSU	22\$		
				0161 31 003D5	28\$: BRW	31\$		1040
		7E	04	A7 3C 003D8	29\$: MOVZWL	4(R7), -(SP)		1249
		7E	02	A7 3C 003DC	MOVZWL	2(R7), -(SP)		
			0573	CB 9F 003E0	PUSHAB	P.AES		
		6A		03 FB 003E4	CALLS	#3, DUMP\$PUT_FAO_		1252
		7E	06	A7 9A 003E7	MOVZBL	6(R7), -(SP)		
		7E	07	A7 9A 003EB	MOVZBL	7(R7), -(SP)		
			059A	CB 9F 003EF	PUSHAB	P.AET		
		6A		03 FB 003F3	CALLS	#3, DUMP\$PUT_FAO_		1254
		50	09	A7 9A 003F6	MOVZBL	9(R7), R0		
	50	50		10 78 003FA	ASHL	#16, R0, R0		
		51	08	A7 9A 003FE	MOVZBL	8(R7), R1		
	7E	50		51 C9 00402	BISL3	R1, R0, -(SP)		
			05C8	CB 9F 00406	PUSHAB	P.AEU		
		6A		02 FB 0040A	CALLS	#2, DUMP\$PUT_FAO_		1255
			05E4	CB 9F 0040D	PUSHAB	P.AEV		
	A0	AA		01 FB 00411	CALLS	#1, DUMP\$FAO_		1256
			44455752	8F DD 00415	PUSHL	#145395026		
		7E	0A	A7 3C 0041B	MOVZWL	10(R7), -(SP)		
	0138	CA		02 FB 0041F	CALLS	#2, FORMAT_PROTECTION		1257
			05FE	CB 9F 00424	PUSHAB	P.AEW		
		A0	AA	01 FB 00428	CALLS	#1, DUMP\$FAO_		1258
				5B DD 0042C	PUSHL	R11		
		7E	0C	A7 3C 0042E	MOVZWL	12(R7), -(SP)		1259
		0B	AA	02 FB 00432	CALLS	#2, FORMAT_BIT_STRING		
				54 DD 00436	PUSHL	R4		
	0175	CA		01 FB 00438	CALLS	#1, FORMAT_RMS_ATTRIBUTES		1260
	00000000G	00		00 FB 0043D	CALLS	#0, DUMP\$B[ANK_LINE		1261
			0630	CB 9F 00444	PUSHAB	P.AEX		
	00000000G	00		01 FB 00448	CALLS	#1, DUMP\$PUT_LINE		1262
			08	AE 9F 0044F	PUSHAB	FILE_NAME		1263
			FA	A2 9F 00452	PUSHAB	-6(IDENT_AREA)		
	00000000G	00		02 FB 00455	CALLS	#2, MAKE_STRING		1266
			08	AE 9F 0045C	PUSHAB	FILE_NAME		
				50 DD 0045F	PUSHL	FILE_LENGTH		
			0638	CB 9F 00461	PUSHAB	P.AEZ		
		6A		03 FB 00465	CALLS	#3, DUMP\$PUT_FAO_		

: 1207 1308 1 END
: 1208 1309 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	8	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	2498	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	3185	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	120 0	1000	00:01.3

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:DUMPHEADE/OBJ=OBJ\$:DUMPHEADE MSRC\$:DUMPHEADE/UPDATE=(ENH\$:DUMPHEADE)

: Size: 2498 code + 3193 data bytes
: Run Time: 00:33.2
: Elapsed Time: 01:55.3
: Lines/CPU Min: 2363
: Lexemes/CPU-Min: 30837
: Memory Used: 454 pages
: Compilation Complete



