


```

IIIIII  NN  NN  SSSSSSSS  KK  KK  FFFFFFFF  SSSSSSSS  CCCCCCCC  AAAAAA  NN  NN
IIIIII  NN  NN  SSSSSSSS  KK  KK  FFFFFFFF  SSSSSSSS  CCCCCCCC  AAAAAA  NN  NN
  II    NN  NN  SS        KK  KK  FF          SS        CC        AA  AA  NN  NN
  II    NN  NN  SS        KK  KK  FF          SS        CC        AA  AA  NN  NN
  II    NNNN  NN  SS        KK  KK  FF          SS        CC        AA  AA  NNNN  NN
  II    NNNN  NN  SS        KK  KK  FF          SS        CC        AA  AA  NNNN  NN
  II    NN  NN  SSSSSS  KKKKKK  FFFFFFFF  SSSSSS  CC        AA  AA  NN  NN
  II    NN  NN  SSSSSS  KKKKKK  FFFFFFFF  SSSSSS  CC        AA  AA  NN  NN
  II    NN  NNNN  SS      KK  KK  FF          SS      CC        AAAAAAAAAA  NN  NNNN
  II    NN  NNNN  SS      KK  KK  FF          SS      CC        AAAAAAAAAA  NN  NNNN
  II    NN  NN  SS      KK  KK  FF          SS      CC        AA  AA  NN  NN
  II    NN  NN  SS      KK  KK  FF          SS      CC        AA  AA  NN  NN
IIIIII  NN  NN  SSSSSSSS  KK  KK  FF          SSSSSSSS  CCCCCCCC  AA  AA  NN  NN
IIIIII  NN  NN  SSSSSSSS  KK  KK  FF          SSSSSSSS  CCCCCCCC  AA  AA  NN  NN

```

```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
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
LLLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS

```

```

1 0001 0 MODULE INSKFSCAN (
2 0002 0
3 0003 0 IDENT = 'V04-000' ; Locate KFE entry
4 0004 0 ADDRESSING_MODE(EXTERNAL = GENERAL)
5 0005 1 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
19 0019 1 * TRANSFERRED. *
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
23 0023 1 * CORPORATION. *
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 **
32 0032 1 FACILITY: Install
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module contains the routine to look up a file in the known
37 0037 1 file list.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1
41 0041 1 VAX/VMS operating system, executive mode.
42 0042 1
43 0043 1 AUTHOR: Bob Grosso, April 1983
44 0044 1
45 0045 1 Modified by:
46 0046 1
47 0047 1 V03-007 RAS0286 Ron Schaefer 9-Apr-1984
48 0048 1 Change normal failure status from 0 to RMSS_FNF so that
49 0049 1 RMS searchlist processing will continue searching.
50 0050 1
51 0051 1 V03-006 MSH0022 Michael S. Harvey 21-Mar-1984
52 0052 1 Correctly message directory specification to eliminate
53 0053 1 middle brackets when root directories are used. This
54 0054 1 ensures that directories as stored in the KFDs are
55 0055 1 correctly matched.
56 0056 1
57 0057 1 V03-005 BLS0256 Benn Schreiber 3-Jan-1984

```

```

: 58      0058 1  Convert square brackets to angle brackets.
: 59      0059 1
: 60      0060 1  V03-004 RPG0004      Bob Grosso      19-Sep-1983
: 61      0061 1  Remove temporary external.
: 62      0062 1
: 63      0063 1  V03-003 RPG0003      Bob Grosso      15-Sep-1983
: 64      0064 1  Replace the hash algorithm.
: 65      0065 1
: 66      0066 1  V03-002 RPG0002      Bob Grosso      July 29, 1983
: 67      0067 1  Add some robustness to my cheesy hash algorithm.
: 68      0068 1  Stuff file id into NAM block for a FID open.
: 69      0069 1
: 70      0070 1  V03-001 RPG0001      Bob Grosso      July 7, 1983
: 71      0071 1  Return special known file found status if file is
: 72      0072 1  installed open.
: 73      0073 1
: 74      0074 1  --
: 75      0075 1
: 76      0076 1
: 77      0077 1  Include files
: 78      0078 1
: 79      0079 1
: 80      0080 1  LIBRARY 'SYSS$LIBRARY:LIB';      ! VAX/VMS system definitions
: 81      0081 1
: 82      0082 1  REQUIRE 'SRC$:INSPREFIX.REQ';
: 83      0224 1  REQUIRE 'LIB$:INSDEF.R32';
```

Declarations

```

: 85      0283 1 %SBTTL 'Declarations';
: 86      0284 1
: 87      0285 1 PSECT DECLARATIONS:
: 88      0286 1
: 89      0287 1
: 90      0288 1 PSECT
: 91      0289 1     CODE    = YF$$$SYSIMGACT (WRITE);
: 92      0290 1
: 93      0291 1
: 94      0292 1 Table of contents
: 95      0293 1
: 96      0294 1 FORWARD ROUTINE
: 97      0295 1     INSSHASH;
: 98      0296 1
: 99      0297 1
: 100     0298 1 EXTERNAL
: 101     0299 1     CTL$GL_KNOWNFIL,           ! Process known file listhead queues
: 102     0300 1     EXE$GL_KNOWN_FILES : REF BBLOCK; ! Pointer to knownfil list queues
: 103     0301 1
: 104     0302 1 LITERAL SGN_B_KFHSHSIZ = 128;
: 105     0303 1

```

```

107 0304 1 %SBTTL 'INSKF_SCAN Locate the KFE';
108 0305 1
109 0306 1 GLOBAL ROUTINE INSKF_SCAN ( KF_NAM, RET_KFE ) =
110 0307 2 BEGIN
111 0308 2 +++
112 0309 2
113 0310 2 FUNCTIONAL DESCRIPTION:
114 0311 2
115 0312 2 EXPLICIT INPUT:
116 0313 2
117 0314 2 IMPLICIT INPUT:
118 0315 2
119 0316 2 IMPLICIT OUTPUT:
120 0317 2
121 0318 2 ROUTINE VALUE:
122 0319 2
123 0320 2 ---
124 0321 2 MAP
125 0322 2 KF_NAM : REF BBLOCK;
126 0323 2
127 0324 2 LOCAL
128 0325 2 DDTSTR : $BBLOCK[NAM$C_MAXRSS],
129 0326 2 DDTSTR_PTR,
130 0327 2 DDTLEN,
131 0328 2 PTR,
132 0329 2 PTR2,
133 0330 2 CMPKFE : REF BBLOCK,
134 0331 2 PRVKFE : REF BBLOCK,
135 0332 2 HSHIDX,
136 0333 2 HSHTAB : REF VECTOR [ ,LONG],
137 0334 2 KFD : REF BBLOCK;
138 0335 2
139 0336 2 BIND
140 0337 2 KFE = .RET_KFE,
141 0338 2 KFPB = EXE$GL_KNOWN_FILES : REF BBLOCK;
142 0339 2
143 0340 2 KFE = 0; ! Assume not found
144 0341 2
145 0342 2 IF .KFPB EQL 0
146 0343 2 THEN
147 0344 2 BEGIN
148 0345 2 |
149 0346 2 | There is no Known file list
150 0347 2 |
151 0348 2 | RETURN RMSS_FNF;
152 0349 2 | END;
153 0350 2
154 0351 2 HSHTAB = .KFPB [KFPB$SL_KFEHSHTAB];
155 0352 2 HSHIDX = INSHASH (.KF_NAM [NAM$B_NAME], .KF_NAM [NAM$L_NAME], SGN_B_KFHSHSIZ);
156 0353 2
157 0354 2 | Search the hash bucket linked list for KFE
158 0355 2 |
159 0356 2 PRVKFE = HSHTAB [.HSHIDX]; ! Previous KFE
160 0357 2 CMPKFE = .HSHTAB [.HSHIDX]; ! Comparison KFE
161 0358 2 WHILE .CMPKFE NEQ 0 DO ! Single linked list ending in zero
162 0359 2 BEGIN
163 0360 2 | CASE CH$COMPARE (.KF_NAM [NAM$B_NAME], .KF_NAM [NAM$L_NAME],

```

164 0361 3
165 0362 3
166 0363 3
167 0364 3
168 0365 3
169 0366 3
170 0367 3
171 0368 3
172 0369 4
173 0370 4
174 0371 4
175 0372 4
176 0373 4
177 0374 4
178 0375 4
179 0376 4
180 0377 4
181 0378 4
182 0379 4
183 0380 4
184 0381 4
185 0382 4
186 0383 4
187 0384 4
188 0385 5
189 0386 5
190 0387 5
191 0388 5
192 0389 5
193 0390 5
194 0391 4
195 0392 4
196 0393 4
197 0394 4
198 0395 4
199 0396 5
200 0397 5
201 0398 5
202 0399 5
203 0400 5
204 0401 5
205 0402 4
206 0403 4
207 0404 4
208 0405 4
209 0406 4
210 0407 4
211 0408 4
212 0409 4
213 0410 4
214 0411 5
215 0412 5
216 0413 5
217 0414 5
218 0415 6
219 0416 6
220 0417 6

```

.CMPKFE [KFESB_FILNAMLEN], CMPKFE [KFEST_FILNAM], %C' ')
FROM -1 TO 1 OF ! Either less than, equal to, or greater than
SET

[-1]: ! Less than, therefore its not in the list
RETURN RMSS_FNF;

[0] : ! Found it
BEGIN
    Build a counted ASCII string of the concatenated Device, Directory
    Type strings.
    DDTLEN = .KF_NAM [NAMSB_DEV] + .KF_NAM [NAMSB_DIR] +
             .KF_NAM [NAMSB_TYPE]; ! Length of DDT string
    DDTSTR_PTR = CH$MOVE (.KF_NAM [NAMSB_DEV], .KF_NAM [NAMSL_DEV], DDTSTR);
    DDTSTR_PTR = CH$MOVE (.KF_NAM [NAMSB_DIR], .KF_NAM [NAMSL_DIR], DDTSTR_PTR);
    DDTSTR_PTR = CH$MOVE (.KF_NAM [NAMSB_TYPE], .KF_NAM [NAMSL_TYPE], DDTSTR_PTR);

    Treat square brackets like angle brackets
    PTR = CH$FIND CH(.DDTLEN,DDTSTR,'[');
    IF NOT CH$FAIC(.PTR)
    THEN
        BEGIN
            CH$WCHAR('<',.PTR);
            PTR = CH$FIND CH(.DDTLEN-(.PTR-DDTSTR),.PTR,'[');
            IF NOT CH$FAIC(.PTR)
            THEN
                CH$WCHAR('<',.PTR);
            END;

        PTR = CH$FIND CH(.DDTLEN,DDTSTR,']');
        IF NOT CH$FAIC(.PTR)
        THEN
            BEGIN
                CH$WCHAR('>',.PTR);
                PTR = CH$FIND CH(.DDTLEN-(.PTR-DDTSTR),.PTR,']');
                IF NOT CH$FAIC(.PTR)
                THEN
                    CH$WCHAR('>',.PTR);
                END;

        Collapse middle directory brackets from root directory

        PTR = CH$FIND CH(.DDTLEN,DDTSTR,'>');
        IF NOT CH$FAIC(.PTR)
        THEN
            BEGIN
                PTR2 = CH$FIND CH(.DDTLEN-(.PTR+1-DDTSTR),.PTR+1,'<');
                IF NOT CH$FAIC(.PTR2)
                THEN
                    BEGIN
                        CH$MOVE(.DDTLEN-(.PTR2+1-DDTSTR),.PTR2+1,.PTR);
                        DDTLEN = .DDTLEN-(.PTR2-.PTR+1);
                    END;
            END;
    
```

```

221 0418 5
222 0419 4
223 0420 4
224 0421 4
225 0422 4
226 0423 4
227 0424 4
228 0425 4
229 0426 4
230 0427 4
231 0428 4
232 0429 4
233 0430 5
234 0431 5
235 0432 5
236 0433 5
237 0434 5
238 0435 5
239 0436 6
240 0437 6
241 0438 6
242 0439 6
243 0440 5
244 0441 5
245 0442 4
246 0443 4
247 0444 5
248 0445 5
249 0446 5
250 0447 4
251 0448 3
252 0449 3
253 0450 3
254 0451 4
255 0452 4
256 0453 4
257 0454 3
258 0455 3
259 0456 2
260 0457 2
261 0458 2
262 0459 2
263 0460 2
264 0461 2
265 0462 2
266 0463 1

```

```

      END;
    END;
  ...
  Traverse the KFD list to find a KFD block with a matching DDT string.
  If no match is found, record address of block after which a new KFD
  block containing the new DDT string should be inserted.
  ...
KFD = .CMPKFE [KFESL KFD];
IF CH$COMPARE (.DDTLEN, DDTSTR,
.KFD [KFD$B_DDTSTRLEN], KFD [KFD$T_DDTSTR], %C' ') EQL 0
THEN
  BEGIN
  KFE = .CMPKFE;
  IF .CMPKFE [KFESV_OPEN]
  THEN
    RETURN RMSS_KFF
  ELSE
    BEGIN
    CH$MOVE (4, CMPKFE [KFESW_FID], KF_NAM [NAMSW_FID]);
    KF_NAM [NAMSW_FID_RVN] = .CMPKFE [RFESW_FID_RVN];
    RETURN TRUE;
    END;
  END
ELSE
  ! Same file name, different file so keep looking
  BEGIN
  PRVKFE = .CMPKFE;
  CMPKFE = .CMPKFE [KFESL_HSHLNK];
  END;
END;
[1] : ! Greater than, keep looking
  BEGIN
  PRVKFE = .CMPKFE;
  CMPKFE = .CMPKFE [KFESL_HSHLNK];
  END;
TES;
END;
      ! WHILE traversing hash bucket list
  ...
  Have traversed whole list without finding it.
  ...
RETURN RMSS_FNF;
END;
      ! Routine INSKF_SCAN

```

```

.TITLE INSKFSCAN
.IDENT \V04-000\

.EXTRN CTL$GL_KNOWNFIL
.EXTRN EXE$GL_KNOWN_FILES

.PSECT YF$$$SYSIMGACT,2

.ENTRY INSKF_SCAN, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0306

```

OFFC 0000

			5E	FEF8	CE	9E	00002	MOVAB	R10, R11		
			08		BC	D4	00007	CLRL	-264(SP), SP		0340
			50	00000000G	00	D0	0000A	MOVL	@RET_KFE		0342
					39	13	00011	BEQL	KFPB, R0		
			52	04	A0	D0	00013	MOVL	3\$		0351
			7E	80	8F	9A	00017	MOVZBL	4(R0), HSHTAB		0352
			57	04	AC	D0	0001B	MOVL	#128, -(SP)		
				4C	A7	DD	0001F	PUSHL	KF NAM, R7		
			7E	38	A7	9A	00022	MOVZBL	76(R7)		
		0000V	CF		03	FB	00026	CALLS	59(R7), -(SP)		
			6E		6240	DE	0002B	MOVAL	#3, INSSHASH		0356
			58		6240	D0	0002F	MOVL	(HSHTAB)[HSHIDX], PRVKFE		0357
					17	13	00033	BEQL	(HSHTAB)[HSHIDX], CMPKFE		0358
			51	38	A7	9A	00035	MOVZBL	3\$		0360
			SU		36	A8	00039	MOVZBL	59(R7), R1		0361
50		20	4C	B7	51	2D	0003D	CMPC5	54(CMPKFE), R0		
					37	A8	00043		R1, @76(R7), #32, R0, 55(CMPKFE)		
					03	1B	00045	BLEQU	2\$		
					0121	31	00047	BRW	16\$		
					03	1E	0004A	BGEQU	4\$		
					0125	31	0004C	BRW	17\$		
			50	39	A7	9A	0004F	MOVZBL	57(R7), R0		0374
			51	3A	A7	9A	00053	MOVZBL	58(R7), R1		
			50		51	C0	00057	ADDL2	R1, R0		
			56	3C	A7	9A	0005A	MOVZBL	60(R7), DDTLEN		0375
			56		50	C0	0005E	ADDL2	R0, DDTLEN		
			50	39	A7	9A	00061	MOVZBL	57(R7), R0		0376
08	AE	44	B7		50	28	00065	MOVC3	R0, @68(R7), DDTSTR		
		04	AE		53	D0	0006B	MOVL	R3, DDTSTR_PTR		
			50	3A	A7	9A	0006F	MOVZBL	58(R7), R0		0377
04	BE	48	B7		50	28	00073	MOVC3	R0, @72(R7), @DDTSTR_PTR		
		04	AE		53	D0	00079	MOVL	R3, DDTSTR_PTR		
			50	3C	A7	9A	0007D	MOVZBL	60(R7), R0		0378
04	BE	50	B7		50	28	00081	MOVC3	R0, @80(R7), @DDTSTR_PTR		
		04	AE		53	D0	00087	MOVL	R3, DDTSTR_PTR		
08	AE		56		5B	8F	0008B	LOCC	#91, DDTLEN, DDTSTR		0382
					02	12	00091	BNEQ	5\$		
					51	D4	00093	CLRL	R1		
			59		51	D0	00095	MOVL	R1, PTR		
					1E	13	00098	BEQL	7\$		0383
			69		3C	90	0009A	MOVB	#60, (PTR)		0386
			50	08	AE	9E	0009D	MOVAB	DDTSTR, R0		0387
			50		59	C2	000A1	SUBL2	PTR, R0		
			50		56	C0	000A4	ADDL2	DDTLEN, R0		
		69	50		5B	8F	000A7	LOCC	#91, R0, (PTR)		
					02	12	000AC	BNEQ	6\$		
					51	D4	000AE	CLRL	R1		
			59		51	D0	000B0	MOVL	R1, PTR		
					03	13	000B3	BEQL	7\$		0388
			69		3C	90	000B5	MOVB	#60, (PTR)		0390
08	AE		56		5D	8F	000B8	LOCC	#93, DDTLEN, DDTSTR		0393
					02	12	000BE	BNEQ	8\$		
					51	D4	000C0	CLRL	R1		
			59		51	D0	000C2	MOVL	R1, PTR		
					1E	13	000C5	BEQL	10\$		0394
			69		3E	90	000C7	MOVB	#62, (PTR)		0397

			50	08	AE	9E	000CA	MOVAB	DDTSTR, R0	0398
			50		59	C2	000CE	SUBL2	PTR, R0	
			50		56	C0	000D1	ADDL2	DDTLEN, R0	
	69		50	5D	8F	3A	000D4	LOCC	#93, R0, (PTR)	
					02	12	000D9	BNEQ	9\$	
					51	D4	000DB	CLRL	R1	
			59		51	D0	000DD	9\$: MOVL	R1, PTR	
					03	13	000E0	BEQL	10\$	0399
	08	AE	69		3E	90	000E2	MOVAB	#62, (PTR)	0401
			56		3E	3A	000E5	10\$: LOCC	#62, DDTLEN, DDTSTR	0408
					02	12	000EA	BNEQ	11\$	
					51	D4	000EC	CLRL	R1	
			59		51	D0	000EE	11\$: MOVL	R1, PTR	
					3D	13	000F1	BEQL	13\$	0409
			50	08	AE	9E	000F3	MOVAB	DDTSTR, R0	0412
			50		59	C2	000F7	SUBL2	PTR, R0	
			50	FF	A046	9E	000FA	MOVAB	-1(R0)[DDTLEN], R0	
			51		59	D0	000FF	MOVL	PTR, R1	
	01	A1	50		3C	3A	00102	LOCC	#60, R0, 1(R1)	
					02	12	00107	BNEQ	12\$	
					51	D4	00109	CLRL	R1	
			5B		51	D0	0010B	12\$: MOVL	R1, PTR2	
					20	13	0010E	BEQL	13\$	0413
			50	08	AE	9E	00110	MOVAB	DDTSTR, R0	0416
			50		5B	C2	00114	SUBL2	PTR2, R0	
			50	FF	A046	9E	00117	MOVAB	-1(R0)[DDTLEN], R0	
			51		5B	D0	0011C	MOVL	PTR2, R1	
	69	01	A1		50	28	0011F	MOVC3	R0, 1(R1), (PTR)	
	51		5B		59	C3	00124	SUBL3	PTR, PTR2, R1	0417
	50		56		51	C3	00128	SUBL3	R1, DDTLEN, R0	
			56	FF	A0	9E	0012C	MOVAB	-1(R0), DDTLEN	
			5A	OC	A8	D0	00130	13\$: MOVL	12(CMPKFE), KFD	0426
			50	10	AA	9A	00134	MOVZBL	16(KFD), R0	0428
			54		01	D0	00138	MOVL	#1, R4	
50		20	08	AE	56	2D	0013B	CMPC5	DDTLEN, DDTSTR, #32, R0, 17(KFD)	
					11	AA	00141			
					03	1A	00143	BGTRU	14\$	
			54		01	D9	00145	SBWC	#1, R4	
					54	D5	00148	14\$: TSTL	R4	
					1F	12	0014A	BNEQ	16\$	
			08	BC	58	D0	0014C	MOVL	CMPKFE, @RET KFE	0431
	08		10	A8	03	E1	00150	BBC	#3, 16(CMPKFE), 15\$	0432
			50	00018031	8F	D0	00155	MOVL	#98353, R0	0436
						04	0015C	RET		
			24	A7	18	A8	0015D	15\$: MOVL	24(CMPKFE), 36(R7)	0437
			28	A7	1C	A8	00162	MOVW	28(CMPKFE), 40(R7)	0438
			50			01	00167	MOVL	#1, R0	0439
						04	0016A	RET		0436
			6E			58	0016B	16\$: MOVL	CMPKFE, PRVKFE	0452
			58			68	0016E	MOVL	(CMPKFE), CMPKFE	0453
						FEBF	31	00171	BRW	1\$
			50	00018292	8F	D0	00174	17\$: MOVL	#98962, R0	0461
						04	0017B	RET		0463

; Routine Size: 380 bytes. Routine Base: YFSSYSIMGACT + 0000

INSKFSCAN
V04-000

: 267

INSSKF_SCAN Locate the KFE

0464 1

1 7
16-Sep-1984 01:58:15
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSKFSCAN.B32;1

Page 9
(3)

IN
VC

```

269 0465 1 %SBTTL 'IN$S$HASH      Compute Hash table index';
270 0466 1
271 0467 1 GLOBAL ROUTINE  IN$S$HASH (LEN, ADR, SIZE) =
272 0468 2 BEGIN
273 0469 2 '+++
274 0470 2
275 0471 2     FUNCTIONAL DESCRIPTION:
276 0472 2
277 0473 2         Return a hash table index computed from the File name
278 0474 2
279 0475 2     EXPLICIT INPUT:
280 0476 2
281 0477 2         LEN      Length of file name
282 0478 2         ADR      Address of name string
283 0479 2
284 0480 2     IMPLICIT INPUT:
285 0481 2
286 0482 2         Assume hash table is 128 longwords
287 0483 2
288 0484 2     ---
289 0485 2 LOCAL
290 0486 2     PTR,
291 0487 2     SUM: BYTE,
292 0488 2     FIRST_CHAR: BYTE;
293 0489 2
294 0490 2     SUM = 0;
295 0491 2     IF .LEN EQL 0 THEN RETURN .SUM;
296 0492 2
297 0493 2     PTR = .ADR;
298 0494 2     INCR INDEX FROM 1 TO .LEN DO
299 0495 3     BEGIN
300 0496 3         SUM = .SUM + .PTR;
301 0497 3         PTR = .PTR + 1;
302 0498 2     END;
303 0499 2
304 0500 2     SUM = .SUM AND 127;
305 0501 2
306 0502 2     RETURN .SUM;
307 0503 1     END;
! Routine IN$S$HASH

```

			0004	00000	.ENTRY	IN\$S\$HASH, Save R2	:	0467
			52	94 00002	CLRB	SUM	:	0490
		04	AC	D5 00004	TSTL	LEN	:	0491
			18	13 00007	BEQL	3\$:	
		51	08	AC D0 00009	MOVL	ADR, PTR	:	0493
			50	D4 0000D	CLRL	INDEX	:	0494
			03	11 0000F	BRB	2\$:	
		52	81	80 00011 1\$:	ADDB2	(PTR)+, SUM	:	0496
	F9	50	04	AC F3 00014 2\$:	AOBLEQ	LEN, INDEX, 1\$:	0494
	52	07	00	EF 00019	EXTZV	#0, #7, SUM, R0	:	0500
		52	50	90 0001E	MOVB	R0, SUM	:	
		50	52	9A 00021 3\$:	MOVZBL	SUM, R0	:	0502
			04	00024	RET		:	0503

INSKFSCAN
V04-000

INSSHASH Compute Hash table index

K 7
16-Sep-1984 01:58:15
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSKFSCAN.B32;1

Page 11
(4)

: Routine Size: 37 bytes, Routine Base: YFSSYSIMGACT + 017C

: 308 0504 1

IN
VC

:
:

INSKFSCAN
V04-000

INSSHASH Compute Hash table index

L 7
16-Sep-1984 01:58:15
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSKFSCAN.B32;1

Page 12
(5)

IN
VO

: 310 0505 1 END ! Module INSKFSCAN
: 311 0506 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
YFSSYSIMGACT	417 NOVEC, WRT, RD ,	EXE,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	31 0	1000	00:01.8

COMMAND QUALIFIERS

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

: Size: 417 code + 0 data bytes
: Run Time: 00:11.1
: Elapsed Time: 00:37.7
: Lines/CPU Min: 2730
: Lexemes/CPU-Min: 15183
: Memory Used: 163 pages
: Compilation Complete

