

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

0001 0 MODULE DIRSCN (LANGUAGE (BLISS32) ,
0002 0             IDENT = 'V04-000'
0003 0             ) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 *  ALL RIGHTS RESERVED.
0011 1 *
0012 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 *  TRANSFERRED.
0018 1 *
0019 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 *  CORPORATION.
0022 1 *
0023 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *
0027 1 *****
0028 1
0029 1 ++
0030 1
0031 1 FACILITY:  F11ACP Structure Level 1
0032 1
0033 1 ABSTRACT:
0034 1
0035 1     This routine performs the basic scan of a directory,
0036 1     searching for both relevant entries and free space.
0037 1 ENVIRONMENT:
0038 1
0039 1     STARLET operating system, including privileged system services
0040 1     and internal exec routines.
0041 1
0042 1 --
0043 1
0044 1
0045 1
0046 1 AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  26-Dec-1976  19:13
0047 1
0048 1 MODIFIED BY:
0049 1
0050 1     V03-004 LMP0281      L. Mark Pilant,      23-Jul-1984  8:43
0051 1     Fix a bug introduced by LMP0255 that caused null directory
0052 1     records to generate the SSS_BADIRECTORY error.
0053 1
0054 1     V03-004 LMP0255      L. Mark Pilant,      6-Jun-1984  10:32
0055 1     Validate the version number from a directory record to
0056 1     insure that it is greater than zero.
0057 1

```

```
58 0058 1 | V03-003 LMP0241 L. Mark Pilant, 26-Apr-1984 16:27  
59 0059 1 | Change the wildcard handling logic to improve performance.  
60 0060 1 |  
61 0061 1 | V03-002 LMP0196 L. Mark Pilant, 28-Feb-1984 11:06  
62 0062 1 | Only use FMG$MATCH_NAME for the file name and type. Not the  
63 0063 1 | version.  
64 0064 1 |  
65 0065 1 | V03-001 LMP0167 L. Mark Pilant, 10-Nov-1983 14:23  
66 0066 1 | Add support for full wildcarding.  
67 0067 1 |  
68 0068 1 | A0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:01  
69 0069 1 | Previous revision history moved to F11A.REV  
70 0070 1 |  
71 0071 1 | **  
72 0072 1 |  
73 0073 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';  
74 0074 1 |  
75 0075 1 | REQUIRE 'SRC$:FCPDEF.B32';  
76 0390 1 |
```

```
78 0391 1 GLOBAL ROUTINE DIR_SCAN (NAME_ OCK, ENTER_MODE) =
79 0392 1
80 0393 1 ++
81 0394 1
82 0395 1 FUNCTIONAL DESCRIPTION:
83 0396 1
84 0397 1 This routine performs the basic scan of a directory,
85 0398 1 searching for both relevant entries and free space.
86 0399 1
87 0400 1 CALLING SEQUENCE:
88 0401 1 DIR_SCAN (ARG1, ARG2)
89 0402 1
90 0403 1 INPUT PARAMETERS:
91 0404 1 ARG1: address of name block
92 0405 1 ARG2: 0 if operation is FIND or REMOVE
93 0406 1 1 if operation is ENTER
94 0407 1
95 0408 1 IMPLICIT INPUTS:
96 0409 1 NONE
97 0410 1
98 0411 1 OUTPUT PARAMETERS:
99 0412 1 NONE
100 0413 1
101 0414 1 IMPLICIT OUTPUTS:
102 0415 1 FIRST FREE: record number of first free directory entry
103 0416 1 DIR_RECORD: record number of found entry
104 0417 1 HIGHEST_VERSION: highest version number encountered
105 0418 1 LOWEST_VERSION: lowest version number encountered
106 0419 1
107 0420 1 ROUTINE VALUE:
108 0421 1 address of found directory entry or
109 0422 1 0 if not found
110 0423 1
111 0424 1 SIDE EFFECTS:
112 0425 1 NONE
113 0426 1
114 0427 1 --
115 0428 1
116 0429 2 BEGIN
117 0430 2
118 0431 2 MAP
119 0432 2 NAME_BLOCK : REF BBLOCK; ! name block arg
120 0433 2
121 0434 2 LINKAGE
122 0435 2 L_MATCH_NAME = JSB (REGISTER = 2, REGISTER = 3,
123 0436 2 REGISTER = 4, REGISTER = 5)
124 0437 2 : NOTUSED (10, 11);
125 0438 2
126 0439 2 EXTERNAL
127 0440 2 FIRST_FREE, ! record number of first free slot
128 0441 2 HIGHEST_VERSION, ! highest version number seen
129 0442 2 LOWEST_VERSION, ! lowest version number seen
130 0443 2 DIR_RECORD; ! record number of found entry
131 0444 2
132 0445 2 EXTERNAL ROUTINE
133 0446 2 DIRGET ! get a directory record
134 0447 2 MAKE_STRING, ! Convert file name block to string
```

```
0448 2          FMGSMATCH_NAME : L_MATCH_NAME;          ! match general wild card string
0449 2
0450 2 LOCAL
0451 2          HIGH_VERSION   : SIGNED WORD,          ! highest version so far
0452 2          LOW_VERSION    : SIGNED WORD,          ! lowest version so far
0453 2          FREE_RECORD    : SIGNED WORD,          ! first free record encountered
0454 2          REC_NUM        : SIGNED WORD,          ! current record number
0455 2          BEST_REC       : SIGNED WORD,          ! record number of highest/lowest version
0456 2          RECADDR        : REF BBLOCK,           ! address of current directory record
0457 2          ASCII_NAME     : VECTOR [20, BYTE];    ! Dir entry name in ASCII
0458 2
0459 2 ! Initialize things.
0460 2
0461 2
0462 2 FREE_RECORD = 0;          ! no empty record found yet
0463 2 BEST_REC = 0;            ! no match found at all
0464 2 HIGH_VERSION = 0;       ! no high version as yet
0465 2 LOW_VERSION = 32767;    ! no low version as yet
0466 2
0467 2 ! Now scan the directory sequentially, looking at each entry. If the search
0468 2 ! is for *.**, then read one block at a time. Otherwise, read whatever the
0469 2 ! buffer pool will take.
0470 2
0471 2
0472 2 REC_NUM = .NAME_BLOCK [NMBSW_CONTEXT];
0473 2 IF
0474 2 BEGIN
0475 2 WHILE 1 DO
0476 2 BEGIN
0477 2 REC_NUM = .REC_NUM + 1;
0478 2 IF .NAME_BLOCK [NMBSV_ALLNAM]
0479 2 AND .NAME_BLOCK [NMBSV_ALLTYP]
0480 2 AND .NAME_BLOCK [NMBSV_ALLVER]
0481 2 THEN RECADDR = DIRGET (.REC_NUM, 0)
0482 2 ELSE RECADDR = DIRGET (.REC_NUM, 1);
0483 2
0484 2 IF .RECADDR EQL 0 THEN EXITLOOP 1;          ! out on end of file
0485 2
0486 2 ! Process each directory entry. First check for empty entries, noting
0487 2 ! the first one.
0488 2
0489 2
0490 2 IF .RECADDR [NMBSW_FID_NUM] EQL 0
0491 2 THEN
0492 2 BEGIN
0493 2 IF .FREE_RECORD EQL 0 THEN FREE_RECORD = .REC_NUM
0494 2 END
0495 2
0496 2 ! If we are in file ID search mode, compare the file ID
0497 2
0498 2
0499 2 ELSE
0500 2 BEGIN
0501 2 IF .RECADDR[NMBSW_VERSION] LEQ 0 THEN ERR_EXIT (SS$_BADIRECTORY);
0502 2 IF .NAME_BLOCK [NMBSV_FINDFID]
0503 2 THEN
0504 2 BEGIN
```

```
192 0505 6      IF CHSEQL (NMB$S_FID, NAME_BLOCK [NMB$W_FID],
193 0506 6          NMB$S_FID, RECADDR [NMB$W_FID], 0)
194 0507 6      THEN
195 0508 6          EXITLOOP 0;
196 0509 6      END
197 0510 6
198 0511 6 ! Otherwise compare the name and type fields under wild card control.
199 0512 6 !
200 0513 6
201 0514 5      ELSE
202 0515 5          IF
203 0516 6              BEGIN
204 0517 6                  IF .NAME_BLOCK[NMB$V_ALLNAM] AND .NAME_BLOCK[NMB$V_ALLTYP]
205 0518 6                  THEN 1
206 0519 6                  ELSE IF .NAME_BLOCK[NMB$V_WILD]
207 0520 6                      THEN
208 0521 7                          BEGIN
209 0522 7                              ASCII_NAME[0] = MAKE_STRING (.RECADDR, ASCII_NAME[1]);
210 0523 7
211 0524 7 ! Note that the match takes place ONLY on the file name and type.
212 0525 7
213 0526 7                  ASCII_NAME[0] = CHSFIND_CH (.ASCII_NAME[0],
214 0527 7                      ASCII_NAME[1], ':') -
215 0528 7                      ASCII_NAME[1];
216 0529 7                  FMG$MATCH_NAME (.ASCII_NAME[0],
217 0530 7                      ASCII_NAME[1],
218 0531 7                      .NAME_BLOCK[NMB$B_ASCNAMSIZ],
219 0532 7                      NAME_BLOCK[NMB$T_ASCNAMTXT])
220 0533 7                  END
221 0534 6              ELSE
222 0535 7                  (CHSEQL (NMB$S_NAME, NAME_BLOCK [NMB$W_NAME],
223 0536 7                      NMB$S_NAME, RECADDR [NMB$W_NAME], 0)
224 0537 7                      AND .NAME_BLOCK [NMB$W_TYPE] EQL .RECADDR [NMB$W_TYPE])
225 0538 6              END
226 0539 6
227 0540 6 ! On a name and type match, process the version. Wild card is an immediate
228 0541 6 ! match. Otherwise, maintain highest and lowest numbers found, keeping
229 0542 6 ! the record number of what we are looking for, if any. On exact match,
230 0543 6 ! exit immediately, note, or ignore, depending on the mode.
231 0544 6 !
232 0545 6
233 0546 5      THEN
234 0547 6          BEGIN
235 0548 6              IF .NAME_BLOCK [NMB$V_ALLVER] THEN EXITLOOP 0;      ! wild card version
236 0549 6
237 0550 6              IF .RECADDR [NMB$W_VERSION] GTR .HIGH_VERSION
238 0551 6              THEN
239 0552 7                  BEGIN
240 0553 7                      HIGH_VERSION = .RECADDR [NMB$W_VERSION];
241 0554 7                      IF NOT .ENTER_MODE AND .NAME_BLOCK [NMB$W_VERSION] EQL 0
242 0555 7                      THEN BEST_REC = .REC_NUM;
243 0556 6                      END;
244 0557 6
245 0558 6              IF .RECADDR [NMB$W_VERSION] LSS .LOW_VERSION
246 0559 6              THEN
247 0560 7                  BEGIN
248 0561 7                      LOW_VERSION = .RECADDR [NMB$W_VERSION];
```

```

: 249 0562 7 IF NOT .ENTER_MODE AND .NAME_BLOCK [NMBSW_VERSION] EQL -32768
: 250 0563 7 THEN
: 251 0564 7 BEST_REC = .REC_NUM;
: 252 0565 6 END;
: 253 0566 6
: 254 0567 6 IF .RECADDR [NMBSW_VERSION] EQL .NAME_BLOCK [NMBSW_VERSION]
: 255 0568 6 THEN
: 256 0569 7 BEGIN
: 257 0570 7 IF NOT .ENTER_MODE THEN EXITLOOP 0;
: 258 0571 7 IF NOT .NAME_BLOCK [NMBSV_NEWVER] THEN BEST_REC = .REC_NUM;
: 259 0572 6 END;
: 260 0573 6
: 261 0574 5 END; ! end of version processing
: 262 0575 4 END; ! end of loop
: 263 0576 4 END END ! end surrounding loop
: 264 0577 3 THEN ! if we scanned the whole directory
: 265 0578 2 ! and found a candidate
: 266 0579 2 IF .BEST_REC NEQ 0
: 267 0580 2 THEN
: 268 0581 3 BEGIN ! re-read the winning entry
: 269 0582 3 RECADDR = DIRGET (.BEST_REC, 0);
: 270 0583 3 REC_NUM = .BEST_REC;
: 271 0584 2 END;
: 272 0585 2
: 273 0586 2 ! Return the implicit outputs as specified and return the address of the
: 274 0587 2 ! found directory entry (or 0 for none).
: 275 0588 2
: 276 0589 2
: 277 0590 2 FIRST_FREE = .FREE_RECORD; ! first free entry
: 278 0591 2 HIGHEST_VERSION = .HIGH_VERSION; ! highest version number found
: 279 0592 2 LOWEST_VERSION = .LOW_VERSION; ! lowest version number found
: 280 0593 2 DIR_RECORD = .REC_NUM; ! record number of entry
: 281 0594 2 RETURN .RECADDR;
: 282 0595 1 END; ! end of routine DIR_SCAN

```

```

.TITLE DIRSCN
.IDENT \V04-000\

.EXTRN FIRST_FREE, HIGHEST_VERSION
.EXTRN LOWEST_VERSION, DIR_RECORD
.EXTRN DIRGET, MAKE_STRING
.EXTRN FMG$MATCH_NAME

```

```
.PSECT $CODE$,NOWRT,2
```

			OFFC 00000	.ENTRY	DIR_SCAN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; R11-	0391
				SUBL2	#28, SP	
		5E	1C C2 00002	CLRQ	BEST_REC	0463
			5A 7C 00005	CLRQ	HIGH_VERSION	0464
		6E	04 AE B4 00007	MOVW	#32767, LOW_VERSION	0465
			7FFF 8F B0 0000A	MOVL	NAME_BLOCK, R0	0472
		50	04 AC D0 0000F	MOVZWL	38(R0), REC_NUM	
		59	26 A0 3C 00013	INCL	REC_NUM	0477
			59 D6 00017 1\$:	MOVL	NAME_BLOCK, R0	0478
		50	04 AC D0 00019	BBC	#5, T6(R0), 2\$	
		0E	10 A0 05 E1 0001D			

09	10	A0	04	E1	00022	BBC	#4, 16(R0), 2\$	0479		
04	10	A0	03	E1	00027	BBC	#3, 16(R0), 2\$	0480		
			7E	D4	0002C	CLRL	-(SP)	0481		
			02	11	0002E	BRB	3\$	0482		
			01	DD	00030	PUSHL	#1	0482		
	0000G	CF	59	DD	00032	PUSHL	REC_NUM			
		57	02	FB	00034	CALLS	#2, DIRGET			
			50	DO	00039	MOVL	R0, RECADDR			
			03	12	0003C	BNEQ	4\$	0484		
			00C	31	0003E	BRW	15\$			
			67	B5	00041	TSTW	(RECADDR)	0490		
			09	12	00043	BNEQ	5\$			
			5B	D5	00045	TSTL	FREE_RECORD	0493		
			CE	12	00047	BNEQ	1\$			
		5B	59	DO	00049	MOVL	REC_NUM, FREE_RECORD			
			C9	11	0004C	BRB	1\$	0492		
			0E	A7	B5	0004E	TSTW	14(RECADDR)	0501	
			0828	05	14	00051	BGTR	6\$		
				8F	BF	00053	CHMU	#2088		
				04	00057	RET				
		56	04	AC	DO	00058	MOVL	NAME_BLOCK, R6	0502	
		58	10	A6	9E	0005C	MOVAB	16(R6), R8		
09		68		0B	E1	00060	BBC	#11, (R8), 7\$		
67		66		06	29	00064	CMPC3	#6, (R6), (RECADDR)	0506	
				AD	12	00068	BNEQ	1\$		
			00B3	31	0006A	BRW	16\$	0508		
		68		05	E1	0006D	BBC	#5, (R8), 8\$	0517	
04		68		04	E0	00071	BBS	#4, (R8), 11\$		
4F		3C		01	A8	E9	00075	BLBC	1(R8), 10\$	0519
				09	AE	9F	00079	PUSHAB	ASCII_NAME+1	0522
					57	DD	0007C	PUSHL	RECADDR	
	0000G	CF		02	FB	0007E	CALLS	#2, MAKE_STRING		
	08	AE		50	90	00083	MOVAB	R0, ASCII_NAME		
09	AE	50	08	AE	9A	00087	MOVZBL	ASCII_NAME, R0	0526	
		50		3B	3A	0008B	LOCC	#59, R0, ASCII_NAME+1	0527	
				02	12	00090	BNEQ	9\$		
				51	D4	00092	CLRL	R1		
08	AE	50	09	AE	9E	00094	MOVAB	ASCII_NAME+1, R0	0528	
		51		50	83	00098	SUBB3	R0, RT, ASCII_NAME		
		55	13	A6	9E	0009D	MOVAB	19(R6), R5	0532	
		53	09	AE	9E	000A1	MOVAB	ASCII_NAME+1, R3	0530	
		54	12	A6	9A	000A5	MOVZBL	18(R6), R4	0532	
		52	08	AE	9A	000A9	MOVZBL	ASCII_NAME, R2		
			0000G	30	000AD	BSBW	FM\$MATCH_NAME			
		57		50	E9	000B0	BLBC	R0, 14\$		
06	A7	06	A6	0F	11	000B3	BRB	11\$		
				06	29	000B5	CMPC3	#6, 6(R6), 6(RECADDR)	0536	
				4D	12	000BB	BNEQ	14\$		
		0C	A7	0C	A6	B1	000BD	CMPL	12(R6), 12(RECADDR)	0537
					46	12	000C2	BNEQ	14\$	
	58	68		03	E0	000C4	BBS	#3, (R8), 16\$	0548	
		50	0E	A7	32	000C8	CVTWL	14(RECADDR), R0	0550	
		50	04	AE	B1	000CC	CMPL	HIGH_VERSION, R0		
				10	18	000D0	BGEQ	12\$		
		04	AE	50	B0	000D2	MOVW	R0, HIGH_VERSION	0553	
		08	08	AC	E8	000D6	BLBS	ENTER_MODE, 12\$	0554	
			0E	A6	B5	000DA	TSTW	14(R6)		

```

03 12 000DD BNEQ 12$
5A 59 D0 000DF MOVL REC_NUM, BEST_REC
50 6E B1 000E2 12$: CMPW LOW_VERSION, R0
12 15 000E5 BLEQ 13$
6E 50 B0 000E7 MOVW R0, LOW_VERSION
08 08 AC E8 000EA BLBS ENTER_MODE, 13$
8000 8F 0E A6 B1 000EE CMPW 14(R6), #-32768
03 03 03 12 000F4 BNEQ 13$
5A 59 D0 000F6 MOVL REC_NUM, BEST_REC
50 0E A6 B1 000F9 13$: CMPW 14(R6), R0
1D 08 AC E9 000FF BLBC ENTER_MODE, 16$
03 68 09 E0 00103 BBS #9, (R8), 14$
5A 59 D0 00107 MOVL REC_NUM, BEST_REC
FF0A 31 0010A 14$: BRW 1$
5A D5 0010D 15$: TSTL BEST_REC
0F 13 0010F BEQL 16$
7E D4 00111 CLRL -(SP)
5A DD 00113 PUSHL BEST_REC
0000G CF 02 FB 00115 CALLS #2, DIRGET
57 50 D0 0011A MOVL R0, RECADDR
59 SA D0 0011D MOVL BEST_REC, REC_NUM
0000G CF 04 5B D0 00120 16$: MOVL FREE_RECORD, FIRST_FREE
0000G CF AE 32 00125 CVTWL HIGH_VERSION, HIGHEST_VERSION
0000G CF 6E 32 0012B CVTWL LOW_VERSION, LOWEST_VERSION
0000G CF 59 D0 00130 MOVL REC_NUM, DIR_RECORD
50 57 D0 00135 MOVL RECADDR, R0
04 00138 RET

```

: Routine Size: 313 bytes. Routine Base: \$CODE\$ + 0000

```

: 283 0596 1
: 284 0597 1 END
: 285 0598 0 ELUDOM

```

PSECT SUMMARY

```

: Name Bytes Attributes
: $CODE$ 313 NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

```

Library Statistics

```

: File Total Symbols Loaded Percent Pages Mapped Processing Time
: _$255$DUA28:[SYSLIB]LIB.L32;1 18619 18 0 1000 00:02.0

```

COMMAND QUALIFIERS

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

: Size: 313 code + 0 data bytes
: Run Time: 00:11.8
: Elapsed Time: 00:29.4
: Lines/CPU Min: 3032
: Lexemes/CPU-Min: 13267
: Memory Used: 160 pages
: Compilation Complete

.....

