

FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFF	111111	111111	AAA	AAA
FFF	111111	111111	AAA	AAA
FFF	111111	111111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111111111	111111111	AAA	AAA
FFF	111111111	111111111	AAA	AAA
FFF	111111111	111111111	AAA	AAA

```

DDDDDDDD      IIIIII      RRRRRRRR      GGGGGGGG      EEEEEEEEEE      TTTTTTTTTT
DDDDDDDD      IIIIII      RRRRRRRR      GGGGGGGG      EEEEEEEEEE      TTTTTTTTTT
DD      DD      II      RR      RR      GG      EE      TT
DD      DD      II      RR      RR      GG      EE      TT
DD      DD      II      RR      RR      GG      EE      TT
DD      DD      II      RRRRRRRR      GG      EEEEEEEE      TT
DD      DD      II      RRRRRRRR      GG      EEEEEEEE      TT
DD      DD      II      RR      RR      GG      GGGGGG      TT
DD      DD      II      RR      RR      GG      GGGGGG      TT
DD      DD      II      RR      RR      GG      GG      TT
DD      DD      II      RR      RR      GG      GG      TT
DD      DD      II      RR      RR      GG      GG      TT
DDDDDDDD      IIIIII      RR      RR      GGGGGG      EEEEEEEEEE      TT
DDDDDDDD      IIIIII      RR      RR      GGGGGG      EEEEEEEEEE      TT

```

```

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
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS

```

.....

```

1 0001 0 MODULE DIRGET (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000',
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
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 |
33 0033 1 | FACILITY: F11ACP Structure Level 1 |
34 0034 1 |
35 0035 1 | ABSTRACT: |
36 0036 1 |
37 0037 1 | This routine gets a directory record from the directory being |
38 0038 1 | operated upon. |
39 0039 1 | ENVIRONMENT: |
40 0040 1 |
41 0041 1 | STARLET operating system, including privileged system services |
42 0042 1 | and internal exec routines. |
43 0043 1 |
44 0044 1 | -- |
45 0045 1 |
46 0046 1 |
47 0047 1 | AUTHOR: Andrew C. Goldstein, CREATION DATE: 26-Dec-1976 18:48 |
48 0048 1 |
49 0049 1 | MODIFIED BY: |
50 0050 1 |
51 0051 1 | A0100 ACG001 Andrew C. Goldstein, 10-Oct-1978 20:01 |
52 0052 1 | Previous revision history moved to F11A.REV |
53 0053 1 |
54 0054 1 |**|
55 0055 1 |
56 0056 1 |
57 0057 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';

```

DIRGET
V04-000

E 4
16-Sep-1984 00:58:53
14-Sep-1984 12:29:29

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]DIRGET.B32;1 Page 2 (1)

: 58 0058 1 REQUIRE 'SRC\$:FCPDEF.B32';

DIRS
V04-

: R

...

...

...

```
60 0373 1 GLOBAL ROUTINE DIRGET (RECNUMBER, MULTI) =
61 0374 1
62 0375 1 !++
63 0376 1
64 0377 1 FUNCTIONAL DESCRIPTION:
65 0378 1
66 0379 1 This routine gets a directory record from the directory being
67 0380 1 operated upon.
68 0381 1
69 0382 1 CALLING SEQUENCE:
70 0383 1 DIRGET (ARG1, ARG2)
71 0384 1
72 0385 1 INPUT PARAMETERS:
73 0386 1 ARG1: record number of directory entry to get
74 0387 1 ARG2: 0 to read just 1 block
75 0388 1 1 to read multiple blocks
76 0389 1
77 0390 1 IMPLICIT INPUTS:
78 0391 1 DIR_FCB: address of FCB for directory
79 0392 1 DIR_WINDOW: address of window for directory
80 0393 1
81 0394 1 OUTPUT PARAMETERS:
82 0395 1 NONE
83 0396 1
84 0397 1 IMPLICIT OUTPUTS:
85 0398 1 NONE
86 0399 1
87 0400 1 ROUTINE VALUE:
88 0401 1 address of record in buffer or
89 0402 1 0 if end of file
90 0403 1
91 0404 1 SIDE EFFECTS:
92 0405 1 directory blocks read into memory
93 0406 1
94 0407 1 --
95 0408 1
96 0409 2 BEGIN
97 0410 2
98 0411 2 LOCAL
99 0412 2 VBN, ! VBN of directory to read
100 0413 2 LBN, ! LBN of directory block
101 0414 2 COUNT, ! number of blocks to read
102 0415 2 ADDRESS; ! buffer address of record
103 0416 2
104 0417 2 EXTERNAL
105 0418 2 DIR_VBN, ! VBN of current directory block
106 0419 2 DIR_BUFFER : REF BBLOCK, ! address of current directory block
107 0420 2 DIR_FCB : REF BBLOCK, ! FCB for directory file
108 0421 2 DIR_WINDOW : REF BBLOCK; ! window for directory file
109 0422 2
110 0423 2 EXTERNAL ROUTINE
111 0424 2 MAP_VBN, ! map VBN to LBN
112 0425 2 READ_BLOCK; ! read a block from the disk
113 0426 2
114 0427 2
115 0428 2 ! Compute the VBN of the directory block containing the desired record.
116 0429 2 ! If the VBN is the same as the last one read, skip the read.
```

```

: 117 0430 2 ! Check it against the directory end of file recorded in the FCB.
: 118 0431 2 !
: 119 0432 2 !
: 120 0433 2 VBN = (.RECNUMBER-1) / (512/16) + 1; ! 512 byte blocks, 16 byte records
: 121 0434 2 IF .VBN NEQ .DIR_VBN
: 122 0435 2 THEN
: 123 0436 2 BEGIN
: 124 0437 2 DIR_VBN = 0;
: 125 0438 2 IF .VBN GTRU .DIR_FCB[FCB$$_EFBLK] THEN RETURN 0; ! no record if beyond end
: 126 0439 2 !
: 127 0440 2 ! Now map VBN to LBN. If the directory is contiguous, use the FCB; if not,
: 128 0441 2 call the mapper with the window. Also, if the directory is contiguous
: 129 0442 2 and multi-block reads are enabled, compute the number of blocks to be read.
: 130 0443 2 !
: 131 0444 2 !
: 132 0445 2 COUNT = 1;
: 133 0446 2 IF .DIR_FCB[FCB$$_STLBN] NEQ 0 ! non-zero start LBN means contiguous
: 134 0447 2 THEN
: 135 0448 2 BEGIN
: 136 0449 2 LBN = .DIR_FCB[FCB$$_STLBN] + .VBN - 1;
: 137 0450 2 IF .MULTI
: 138 0451 2 THEN COUNT = .DIR_FCB[FCB$$_EFBLK] - .VBN + 1;
: 139 0452 2 END
: 140 0453 2 ELSE
: 141 0454 2 BEGIN
: 142 0455 2 LBN = MAP_VBN (.VBN, .DIR_WINDOW);
: 143 0456 2 IF .LBN EQL -1 THEN RETURN 0; ! EOF if map fails
: 144 0457 2 END;
: 145 0458 2 !
: 146 0459 2 ! Read the block and point to the record within it.
: 147 0460 2 !
: 148 0461 2 !
: 149 0462 2 DIR_BUFFER = READ_BLOCK (.LBN, .COUNT, DIRECTORY_TYPE);
: 150 0463 2 DIR_VBN = .VBN;
: 151 0464 2 END;
: 152 0465 2 !
: 153 0466 2 ADDRESS = .DIR_BUFFER + (.RECNUMBER-1)*16 - (.VBN-1)*512;
: 154 0467 2 RETURN .ADDRESS;
: 155 0468 2 !
: 156 0469 1 END; ! end of routine DIRGET

```

```

.TITLE DIRGET
.IDENT \V04-000\

.EXTRN DIR_VBN, DIR_BUFFER
.EXTRN DIR_FCB, DIR_WINDOW
.EXTRN MAP_VBN, READ_BLOCK

.PSECT $CODE$,NOWRT,2

.ENTRY DIRGET, Save R2,R3,R4,R5
MOVAB DIR_VBN, R5
SUBL3 #1, RECNUMBER, R4
DIVL3 #32, R4, R0
MOVAB 1(R0), VBN
CMPL VBN, DIR_VBN

```

```

54 04 55 0000G 003C 00000
50 04 AC 01 C3 00002
54 20 C7 0000C
52 01 A0 9E 00010
65 52 D1 00014

```

```

: 0373
: 0433
: 0434

```

			53	13	00017	BEQL	3\$		
			65	D4	00019	CLRL	DIR_VBN		0437
	3C	50	CF	D0	0001B	MOVL	DIR_FCB, R0		0438
		A0	52	D1	00020	CML	VBN, 60(R0)		
			5C	1A	00024	BGTRU	4\$		
		53	01	D0	00026	MOVL	#1, COUNT		0445
			30	A0	D5 00029	TSTL	48(R0)		0446
51		52	30	A0	C1 0002E	BEQL	1\$		
				51	D7 00033	ADDL3	48(R0), VBN, R1		0449
		22	08	AC	E9 00035	DECL	LBN		
50	3C	A0	52	C3	00039	BLBC	MULTI, 2\$		0450
		53	01	A0	9E 0003E	SUBL3	VBN, 60(R0) R0		0451
				17	11 00042	MOVAB	1(R0), COUNT		
			0000G	CF	DD 00044	BRB	2\$		0446
				52	DD 00048	PUSHL	DIR_WINDOW		0455
	0000G	CF		02	FB 0004A	PUSHL	VBN		
	FFFFFFF	51		50	D0 0004F	CALLS	#2, MAP_VBN		
		8F		51	D1 00052	MOVL	R0, LBN		
				27	13 00059	CML	LBN, #-1		0456
				02	DD 0005B	BEQL	4\$		
				0A	BB 0005D	PUSHL	#2		0462
	0000G	CF		03	FB 0005F	PUSHR	#^M<R1,R3>		
	0000G	CF		50	D0 00064	CALLS	#3, READ_BLOCK		
		65		52	D0 00069	MOVL	R0, DIR_BUFFER		
50		54		04	78 0006C	MOVL	VBN, DIR_VBN		0463
		50		CF	C0 00070	ASHL	#4, R4, R0		0466
52		52	0000G	09	78 00075	ADDL2	DIR_BUFFER, R0		
		50		52	C2 00079	ASHL	#9, R2, R2		
		50		CO	9E 0007C	SUBL2	R2, R0		
			0200	CO	04 00081	MOVAB	512(R0), ADDRESS		0467
				50	D4 00082	RET			0469
				04	00084	CLRL	R0		
						RET			

; Routine Size: 133 bytes, Routine Base: \$CODE\$ + 0000

```

: 157      0470 1
: 158      0471 1 END
: 159      0472 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	133	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

DIRGET
V04-000

1 4
16-Sep-1984 00:58:53
14-Sep-1984 12:29:29

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]DIRGET.B32;1 Page 6
(2)

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
:_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	3	0	1000	00:02.0

COMMAND QUALIFIERS

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

: Size: 133 code + 0 data bytes
: Run Time: 00:06.7
: Elapsed Time: 00:20.7
: Lines/CPU Min: 4220
: Lexemes/CPU-Min: 10819
: Memory Used: 82 pages
: Compilation Complete

0165 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

A grid of 15 columns and 15 rows of small, illegible text screens, likely representing a terminal window or a list of system components. The screens are arranged in a regular pattern across the page.

EXTFCB
LIS

DELJL
LIS

DIRGET
LIS

EXTIOX
LIS

IODONE
LIS

LOCKDN
LIS

ENTER
LIS

GETREQ
LIS

GETTIM
LIS

DISPAT
LIS

INIFCP
LIS

DIRFCB
LIS

EXTHDR
LIS

DIRSCN
LIS

LOGDEL
LIS

LOCKDB
LIS

DIRACC
LIS

FIND
LIS

GETFIB
LIS

INIFCB
LIS

EXTDIR
LIS

EXTEND
LIS