

```
FFFFFFFFFFFFFFFF 111 111 111
FFFFFFFFFFFFFFFF 111 111 111
FFFFFFFFFFFFFFFF 111 111 111
FFF 111111 111111 111111
FFF 111111 111111 111111
FFF 111111 111111 111111
FFF 111 111 111
FFF 111 111 111
FFF 111 111 111
FFFFFFFFFFFFFF 111 111 111
FFFFFFFFFFFFFF 111 111 111
FFFFFFFFFFFFFF 111 111 111
FFF 111 111 111
FFF 111 111 111
FFF 111 111 111
FFF 111 111 111
FFF 111 111 111
FFF 111 111 111
FFF 111 111 111
FFF 111111111 111111111 111111111
FFF 111111111 111111111 111111111
FFF 111111111 111111111 111111111

AAAAAAAAAA AAA AAA
AAAAAAAAAA AAA AAA
AAAAAAAAAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
AAA AAA AAA
```



```

1 0001 0 MODULE CREWIN (
2 0002 0
3 0003 0 LANGUAGE (BLISS32),
4 0004 0 IDENT = 'V04-000'
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 creates and initializes a file window.
38 0038 1 ENVIRONMENT:
39 0039 1
40 0040 1 STARLET operating system, including privileged system services
41 0041 1 and internal exec routines. This routine must be called
42 0042 1 in kernel mode.
43 0043 1
44 0044 1 --
45 0045 1
46 0046 1
47 0047 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 14-Dec-1976 17:10
48 0048 1
49 0049 1 MODIFIED BY:
50 0050 1
51 0051 1 V03-001 LMP0018 L. Mark Pilant, 31-Mar-1982 13:13
52 0052 1 Modify to use a local of the window complete flag.
53 0053 1
54 0054 1 V02-001 LMP0005 L. Mark Pilant, 29-Dec-1981 14:40
55 0055 1 Add support for Cathedral windows.
56 0056 1
57 0057 1 A0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:01

```

CREWIN
V04-000

L 15
16-Sep-1984 00:55:27
14-Sep-1984 12:29:26

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

Previous revision history moved to F11A.REV

:	58	0058	1	:	
:	59	0059	1	:	
:	60	0060	1	!*	
:	61	0061	1	:	
:	62	0062	1	:	
:	63	0063	1	LIBRARY	'SYSSLIBRARY:LIB.L32';
:	64	0064	1	REQUIRE	'SRC\$:FCPDEF.B32';

```

: 66 0379 1 GLOBAL ROUTINE CREATE_WINDOW (ACCTL, SIZE, HEADER, PID, FCB) =
: 67 0380 1
: 68 0381 1 !++
: 69 0382 1
: 70 0383 1 FUNCTIONAL DESCRIPTION:
: 71 0384 1
: 72 0385 1 This routine creates a file access window.
: 73 0386 1
: 74 0387 1 CALLING SEQUENCE:
: 75 0388 1 CREATE_WINDOW (ARG1, ARG2, ARG3, ARG4, ARG5)
: 76 0389 1
: 77 0390 1 INPUT PARAMETERS:
: 78 0391 1 ARG1: access control word (from FIB, usually)
: 79 0392 1 ARG2: size of window in # of pointers
: 80 0393 1 ARG3: address of file header
: 81 0394 1 ARG4: PID of accessor
: 82 0395 1 ARG5: address of file FCB
: 83 0396 1
: 84 0397 1 IMPLICIT INPUTS:
: 85 0398 1 CURRENT_VCB: address of VCB of volume in process
: 86 0399 1 CURRENT_UCB: address of UCB of disk in process
: 87 0400 1
: 88 0401 1 OUTPUT PARAMETERS:
: 89 0402 1 NONE
: 90 0403 1
: 91 0404 1 IMPLICIT OUTPUTS:
: 92 0405 1 NONE
: 93 0406 1
: 94 0407 1 ROUTINE VALUE:
: 95 0408 1 address of window
: 96 0409 1
: 97 0410 1 SIDE EFFECTS:
: 98 0411 1 window block created
: 99 0412 1
: 100 0413 1 --
: 101 0414 1
: 102 0415 2 BEGIN
: 103 0416 2
: 104 0417 2 MAP
: 105 0418 2 HEADER : REF BBLOCK, ! file header arg
: 106 0419 2 FCB : REF BBLOCK; ! FCB arg
: 107 0420 2
: 108 0421 2 LOCAL
: 109 0422 2 WINDOW_SIZE, ! actual size of window
: 110 0423 2 WINDOW : REF BBLOCK, ! window created
: 111 0424 2 PRIMARY_WINDOW : REF BBLOCK; ! address of the primary window
: 112 0425 2
: 113 0426 2 EXTERNAL
: 114 0427 2 CURRENT_VCB : REF BBLOCK, ! VCB in process
: 115 0428 2 CURRENT_UCB : REF BBLOCK; ! UCB in process
: 116 0429 2
: 117 0430 2 EXTERNAL ROUTINE
: 118 0431 2 ALLOCATE, ! allocate dynamic memory
: 119 0432 2 TURN_WINDOW; ! window turner routine
: 120 0433 2
: 121 0434 2 ! Compute the size of the window. If fixed, allocate it and turn it to
: 122 0435 2 ! map VBN 1. If a maximal window is requested (indicated by a size of -1),

```

```

: 123 0436 2 ! the window turner will allocate the window.
: 124 0437 2 !
: 125 0438 2
: 126 0439 WINDOW SIZE = .SIZE;
: 127 0440 IF .WINDOW SIZE EQL 0
: 128 0441 THEN WINDOW_SIZE = .CURRENT_VCB[VCB$B_WINDOW];
: 129 0442
: 130 0443 IF .WINDOW_SIZE NEQ -1
: 131 0444 THEN
: 132 0445 BEGIN
: 133 0446 IF .WINDOW SIZE GTRU MAX WINDOW
: 134 0447 THEN WINDOW_SIZE = MAX WINDOW;
: 135 0448 IF .WINDOW SIZE LSSU MIN WINDOW
: 136 0449 THEN WINDOW_SIZE = MIN WINDOW;
: 137 0450 WINDOW = ALLOCATE (.WINDOW_SIZE * 6 + WCB$C_LENGTH, WCB_TYPE);
: 138 0451 IF .WINDOW NEQ 0
: 139 0452 THEN TURN_WINDOW (.WINDOW, .HEADER, 1, 1);
: 140 0453 END
: 141 0454 ELSE
: 142 0455 WINDOW = TURN_WINDOW (0, .HEADER, 1, 1);
: 143 0456
: 144 0457 ! Init cells within the window
: 145 0458 !
: 146 0459
: 147 0460 PRIMARY WINDOW = .WINDOW;
: 148 0461 UNTIL .WINDOW EQL 0
: 149 0462 DO
: 150 0463 BEGIN
: 151 0464 WINDOW[WCB$S_PID] = .PID; ! accessor PID
: 152 0465 WINDOW[WCB$S_ORGUCB] = .CURRENT_UCB; ! original device UCB
: 153 0466 WINDOW[WCB$S_ACON] = .ACCTL<0,16>; ! access control bits
: 154 0467 WINDOW[WCB$S_FCB] = .FCB; ! FCB address
: 155 0468 WINDOW[WCB$S_READ] = 1; ! read access always allowed
: 156 0469 WINDOW[WCB$S_WRITE] = .WINDOW[WCB$S_WRITEAC]; ! write access sometimes
: 157 0470 IF .HEADER[FH1$V_READCHECK] THEN WINDOW[WCB$S_READCK] = 1;
: 158 0471 IF .HEADER[FH1$V_WRITECHECK] THEN WINDOW[WCB$S_WRITECK] = 1;
: 159 0472 WINDOW = .WINDOW[WCB$S_LINK];
: 160 0473 END;
: 161 0474
: 162 0475 RETURN .PRIMARY_WINDOW;
: 163 0476
: 164 0477 1 END;

```

! end of routine CREATE_WINDOW

```

.TITLE CREWIN
.IDENT \V04-000\

.EXTRN CURRENT_VCB, CURRENT_UCB
.EXTRN ALLOCATE, TURN_WINDOW

.PSECT $CODE$,NOWRT,2

.ENTRY CREATE_WINDOW, Save R2
MOVL SIZE, WINDOW_SIZE
BNEQ 1$
MOVL CURRENT_VCB, R1
MOVZBL 72(R1), WINDOW_SIZE

```

```

0004 00000
50 08 AC D0 00002
09 12 00006
51 0000G CF D0 00008
50 48 A1 9A 0000D

```

```

: 0379
: 0439
: 0440
: 0441
:

```

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

FFFFFFF	8F		50	D1	00011	1\$:	CMPL	WINDOW_SIZE, #-1	: 0443
			36	13	00018		BEQL	4\$: 0444
0000050	8F		50	D1	0001A		CMPL	WINDOW_SIZE, #80	: 0446
			04	1B	00021		BLEQU	2\$: 0447
		50	50	8F	9A	00023	MOVZBL	#80, WINDOW_SIZE	: 0448
			50	D5	00027	2\$:	TSTL	WINDOW_SIZE	: 0449
			03	12	00029		BNEQ	3\$: 0450
		50	01	D0	0002B		MOVL	#1, WINDOW_SIZE	: 0451
		50	01	DD	0002E	3\$:	PUSHL	#1	: 0452
			06	C4	00030		MULL2	#6, R0	: 0453
0000G	CF		30	A0	9F	00033	PUSHAB	48(R0)	: 0454
	52		02	FB	00036		CALLS	#2, ALLOCATE	: 0455
			50	D0	0003B		MOVL	R0, WINDOW	: 0456
			21	13	0003E		BEQL	5\$: 0457
			01	DD	00040		PUSHL	#1	: 0458
			01	DD	00042		PUSHL	#1	: 0459
			0C	AC	DD	00044	PUSHL	HEADER	: 0460
0000G	CF		52	DD	00047		PUSHL	WINDOW	: 0461
			04	FB	00049		CALLS	#4, TURN_WINDOW	: 0462
			11	11	0004E		BRB	5\$: 0463
			01	DD	00050	4\$:	PUSHL	#1	: 0464
			01	DD	00052		PUSHL	#1	: 0465
			0C	AC	DD	00054	PUSHL	HEADER	: 0466
0000G	CF		7E	D4	00057		CLRL	-(SP)	: 0467
	52		04	FB	00059		CALLS	#4, TURN_WINDOW	: 0468
	51		50	D0	0005E		MOVL	R0, WINDOW	: 0469
	50		52	D0	00061	5\$:	MOVL	WINDOW, PRIMARY_WINDOW	: 0470
			0C	AC	D0	00064	MOVL	HEADER, R0	: 0471
			52	D5	00068	6\$:	TSTL	WINDOW	: 0472
			38	13	0006A		BEQL	9\$: 0473
	0C	A2	10	AC	D0	0006C	MOVL	PID, 12(WINDOW)	: 0474
	10	A2	0000G	CF	D0	00071	MOVL	CURRENT_UCB, 16(WINDOW)	: 0475
	14	A2	04	AC	B0	00077	MOVW	ACCTL, 20(WINDOW)	: 0476
	18	A2	14	AC	D0	0007C	MOVL	FCB, 24(WINDOW)	: 0477
	OB	A2	01	88	00081		BISB2	#1, 11(WINDOW)	: 0478
OB	A2	01	15	A2	F0	00085	INSV	21(WINDOW), #1, #1, 11(WINDOW)	: 0479
	04	0C	03	E1	0008C		BBC	#3, 12(R0), 7\$: 0480
	15	A2	02	88	00091		BISB2	#2, 21(WINDOW)	: 0481
	04	0C	04	E1	00095	7\$:	BBC	#4, 12(R0), 8\$: 0482
	14	A2	20	88	0009A		BISB2	#32, 20(WINDOW)	: 0483
		52	20	A2	D0	0009E	MOVL	32(WINDOW), WINDOW	: 0484
				C4	11	000A2	BRB	6\$: 0485
		50	51	D0	000A4	9\$:	MOVL	PRIMARY_WINDOW, R0	: 0486
			04	000A7			RET		: 0487

; Routine Size: 168 bytes, Routine Base: \$CODE\$ + 0000

```

: 165      0478 1
: 166      0479 1 END
: 167      0480 0 ELUDOM

```

PSECT SUMMARY

```
:  
: Name Bytes Attributes  
: $CODE$ 168 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 15 0 1000 00:02.0
```

COMMAND QUALIFIERS

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

```
: Size: 168 code + 0 data bytes  
: Run Time: 00:07.2  
: Elapsed Time: 00:23.2  
: Lines/CPU Min: 4016  
: Lexemes/CPU-Min: 12786  
: Memory Used: 98 pages  
: Compilation Complete
```


ACCPNTR LIS	CHKSUM LIS	CHKPRO LIS	DEACCS LIS
FCPDEF B32	BADSEN LIS	CLENUP LIS	CPYNAM LIS
CHKHDR LIS	COMMON LIS	CREHDR LIS	CREWIN LIS
ALLOB LIS	ACCESS LIS	CHKDMD LIS	DELETE LIS
CREATE LIS	CREFCB LIS		