


```

CCCCCCCC 000000 MM MM MM MM 000000 NN NN
CCCCCCCC 000000 MM MM MM MM 000000 NN NN
CC        00      00 MMMM MMMM MMMM MMMM 00      00 NN NN NN
CC        00      00 MMMM MMMM MMMM MMMM 00      00 NN NN NN
CC        00      00 MM MM MM MM MM MM 00      00 NNNN NN
CC        00      00 MM MM MM MM MM MM 00      00 NNNN NN
CC        00      00 MM MM MM MM MM MM 00      00 NN NN NN
CC        00      00 MM MM MM MM MM MM 00      00 NN NN NN
CC        00      00 MM MM MM MM MM MM 00      00 NN NN NN
CC        00      00 MM MM MM MM MM MM 00      00 NN NN NN
CC        00      00 MM MM MM MM MM MM 00      00 NN NN NN
CC        00      00 MM MM MM MM MM MM 00      00 NN NN NN
CCCCCCCC 000000 MM MM MM MM 000000 NN NN
CCCCCCCC 000000 MM MM MM MM 000000 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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```

```

....
....
....
....

```



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 COMMON (
0002 0
0003 0     LANGUAGE (BLISS32),
0004 0     IDENT = 'V04-000'
0005 1 ) =
0006 1 BEGIN
0007 1
0008 1 *****
0009 1 *
0010 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 *  ALL RIGHTS RESERVED.
0013 1 *
0014 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 *  TRANSFERRED.
0020 1 *
0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 *  CORPORATION.
0024 1 *
0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 ++
0032 1
0033 1 FACILITY: F11ACP Structure Level 1
0034 1
0035 1 ABSTRACT:
0036 1
0037 1     This module is the common impure area of FCP, including the
0038 1     routine that initializes it for each request.
0039 1
0040 1 ENVIRONMENT:
0041 1
0042 1     STARLET operating system, including privileged system services
0043 1     and internal exec routines.
0044 1
0045 1 --
0046 1
0047 1
0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 23:42
0049 1
0050 1 MODIFIED BY:
0051 1
0052 1     V03-002 ACG0408 Andrew C. Goldstein, 20-Mar-1984 16:36
0053 1     Reduce size of LOCAL_ARB
0054 1
0055 1     V03-001 ACG0346 Andrew C. Goldstein, 2-Aug-1983 15:15
0056 1     Remove job controller mailbox cells
0057 1

```

```

: 58      0058 1  : V02-003 ACG0245      Andrew C. Goldstein, 22-Dec-1981 20:18
: 59      0059 1  :           Add job controller mailbox channel
: 60      0060 1  :
: 61      0061 1  : V02-002 ACG38100     Andrew C. Goldstein, 3-Jun-1981 11:50
: 62      0062 1  :           Fix granting of SYSPRV to volume owner
: 63      0063 1  :
: 64      0064 1  : V02-001 ACG0167     Andrew C. Goldstein, 7-May-1980 18:48
: 65      0065 1  :           Previous revision history moved to f11A.REV
: 66      0066 1  : **
: 67      0067 1  :
: 68      0068 1  :
: 69      0069 1  : LIBRARY 'SYSS$LIBRARY:LIB;L32';
: 70      0070 1  : REQUIRE 'SRC$:FCPDEF.B32';
: 71      0385 1  :
: 72      0386 1  :
: 73      0387 1  : FORWARD ROUTINE
: 74      0388 1  :     INIT_COMMON      : NOVALUE,      ! initialize global storage
: 75      0389 1  :     SAVE_CONTEXT    : NOVALUE,      ! save reentrant context area
: 76      0390 1  :     RESTORE_CONTEXT : NOVALUE;      ! restore reentrant context area

```

```

78 0391 1 | |
79 0392 1 | |
80 0393 1 | | ACP global impure area
81 0394 1 | |
82 0395 1 | |
83 0396 1 | |
84 0397 1 | GLOBAL
85 0398 1 |     QUEUE_HEAD      : REF BBLOCK,  ! pointer to ACP queue header
86 0399 1 |     IO_CHANNEL,    : REF BBLOCK,  ! channel number for all I/O
87 0400 1 |     DISK_UCB       : REF BBLOCK,  ! original UCB of our channel
88 0401 1 |
89 0402 1 | | The remaining locations are initialized to known values (mainly zero)
90 0403 1 | | by the routine.
91 0404 1 | |
92 0405 1 |     USER_STATUS    : VECTOR [2],  ! I/O status to be returned to user
93 0406 1 |     IO_STATUS      : VECTOR [2],  ! status block for FCP I/O
94 0407 1 |     IO_PACKET      : REF BBLOCK,  ! address of current I/O request packet
95 0408 1 |     CURRENT_UCB    : REF BBLOCK,  ! address of UCB of current request
96 0409 1 |     CURRENT_VCB    : REF BBLOCK,  ! address of VCB of current request
97 0410 1 |     NEW_FID,       : REF BBLOCK,  ! file number of unrecorded file ID
98 0411 1 |     HEADER_LBN,    : REF BBLOCK,  ! LBN of last file header read
99 0412 1 |     DIR_FCB        : REF BBLOCK,  ! FCB of directory file
100 0413 1 |     DIR_WINDOW     : REF BBLOCK,  ! window of current directory
101 0414 1 |     BITMAP_VBN,    : REF BBLOCK,  ! VBN of current storage map block
102 0415 1 |     BITMAP_BUFFER   : REF BBLOCK,  ! address of current storage map block
103 0416 1 |     DIR_VBN        : REF BBLOCK,  ! VBN of current directory block
104 0417 1 |     DIR_BUFFER     : REF BBLOCK,  ! address of current directory block
105 0418 1 |     DIR_RECORD,    : REF BBLOCK,  ! record number of found directory entry
106 0419 1 |     HIGHEST_VERSION, : REF BBLOCK,  ! highest version number in directory
107 0420 1 |     LOWEST_VERSION, : REF BBLOCK,  ! lowest version number in directory
108 0421 1 |     FIRST_FREE,    : REF BBLOCK,  ! record number of first free entry
109 0422 1 |     SUPER_FID      : BBLOCK [FID$C_LENGTH], ! file ID of superseded file
110 0423 1 |     LOCAL_FIB      : BBLOCK [FIB$C_LENGTH], ! primary FIB of this operation
111 0424 1 |     SECOND_FIB     : BBLOCK [FIB$C_LENGTH], ! FIB for secondary file operation
112 0425 1 |     LOCAL_ARB      : BBLOCK [ARB$C_HEADER], ! local copy of caller's ARB
113 0426 1 |
114 0427 1 | | The following locations are the re-entertainable
115 0428 1 | | context area and must be saved when an
116 0429 1 | | secondary operation is performed.
117 0430 1 |     CONTEXT_START  : VECTOR [0],  ! **** The next item must be CLEANUP_FLAGS
118 0431 1 |     CLEANUP_FLAGS  : BITVECTOR [32], ! cleanup action flags
119 0432 1 |     FILE_HEADER    : REF BBLOCK,  ! address of current file header
120 0433 1 |     UNREC_LBN,     : REF BBLOCK,  ! start LBN of unrecorded blocks
121 0434 1 |     UNREC_COUNT,   : REF BBLOCK,  ! count of unrecorded blocks
122 0435 1 |     PRIMARY_FCB    : REF BBLOCK,  ! address of primary file FCB
123 0436 1 |     CURRENT_WINDOW : REF BBLOCK,  ! address of file window
124 0437 1 |     CURRENT_FIB    : REF BBLOCK,  ! pointer to FIB currently in use
125 0438 1 |     CONTEXT_END    : VECTOR [0];
126 0439 1 |
127 0440 1 | GLOBAL LITERAL
128 0441 1 |     CONTEXT_SIZE   = CONTEXT_END - CONTEXT_START; ! byte count of context area
129 0442 1 |
130 0443 1 | GLOBAL
131 0444 1 |     CONTEXT_SAVE   : VECTOR [CONTEXT_SIZE, BYTE], ! area to save primary context
132 0445 1 |
133 0446 1 |     IMPURE_END     : VECTOR [0]; ! end of impure area
134 0447 1 |

```

COMMON
V04-000

M 11
16-Sep-1984 00:52:09
14-Sep-1984 12:29:23

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[F11A.SRC]COMMON.B32;1 Page 4 (2)

: 135
: 136

0448 1 GLOBAL LITERAL
0449 1 IMPURE_SIZE = IMPURE_END - USER_STATUS; ! byte count of impure area

CPY
V04
:
: : : : :
S
R
J
E
C

```

: 138 0450 1 GLOBAL ROUTINE INIT_COMMON : NOVALUE =
: 139 0451 1
: 140 0452 1 !++
: 141 0453 1
: 142 0454 1 FUNCTIONAL DESCRIPTION:
: 143 0455 1
: 144 0456 1 This routine contains the impure data base for FCP, and is called
: 145 0457 1 to initialize it.
: 146 0458 1
: 147 0459 1 CALLING SEQUENCE:
: 148 0460 1 INIT_COMMON ()
: 149 0461 1
: 150 0462 1 INPUT PARAMETERS:
: 151 0463 1 NONE
: 152 0464 1
: 153 0465 1 IMPLICIT INPUTS:
: 154 0466 1 NONE
: 155 0467 1
: 156 0468 1 OUTPUT PARAMETERS:
: 157 0469 1 NONE
: 158 0470 1
: 159 0471 1 IMPLICIT OUTPUTS:
: 160 0472 1 NONE
: 161 0473 1
: 162 0474 1 ROUTINE VALUE:
: 163 0475 1 NONE
: 164 0476 1
: 165 0477 1 SIDE EFFECTS:
: 166 0478 1 DATABASE INITIALIZED
: 167 0479 1
: 168 0480 1 !--
: 169 0481 1
: 170 0482 2 BEGIN
: 171 0483 2
: 172 0484 2 ! Initialization consists of zeroing the impure area and then setting the
: 173 0485 2 ! user request status to 1 (success).
: 174 0486 2 !
: 175 0487 2
: 176 0488 2 CH$FILL (0, IMPURE_SIZE, USER_STATUS);
: 177 0489 2 USER_STATUS[0] = 1;
: 178 0490 2
: 179 0491 1 END;

```

! end of routine INIT_COMMON

```

.TITLE COMMON
.IDENT \V04-000\
.PSECT $LOCKEDD1$,NOEXE,2

```

```

00000 QUEUE_HEAD::
         .BLKB 4
00004 IO_CHANNEL::
         .BLKB 4
00008 DISK_UCB::
         .BLKB 4
0000C USER_STATUS::
         .BLKB 8

```

00014	IO_STATUS::	.BLKB	8
0001C	IO_PACKET::	.BLKB	4
00020	CURRENT_UCB::	.BLKB	4
00024	CURRENT_VCB::	.BLKB	4
00028	NEW_FID::	.BLKB	4
0002C	HEADER_LBN::	.BLKB	4
00030	DIR_FCB::	.BLKB	4
00034	DIR_WINDOW::	.BLKB	4
00038	BITMAP_VBN::	.BLKB	4
0003C	BITMAP_BUFFER::	.BLKB	4
00040	DIR_VBN::	.BLKB	4
00044	DIR_BUFFER::	.BLKB	4
00048	DIR_RECORD::	.BLKB	4
0004C	HIGHEST_VERSION::	.BLKB	4
00050	LOWEST_VERSION::	.BLKB	4
00054	FIRST_FREE::	.BLKB	4
00058	SUPER_FID::	.BLKB	6
0005E	LOCAL_FIB::	.BLKB	2
00060	SECOND_FIB::	.BLKB	64
000A0	LOCAL_ARB::	.BLKB	64
000E0	CONTEXT_START::	.BLKB	48
00110	CLEANUP_FLAGS::	.BLKB	0
00114	FILE_HEADER::	.BLKB	4
00118	UNREC_LBN::	.BLKB	4
0011C	UNREC_COUNT::	.BLKB	4
00120	PRIMARY_FCB::	.BLKB	4
00124	CURRENT_WINDOW::	.BLKB	4
00128	CURRENT_FIB::	.BLKB	4

0012C CONTEXT_END::
 .BLKB 0
0012C CONTEXT_SAVE::
 .BLKB 28
00148 IMPURE_END::
 .BLKB 0

CONTEXT_SIZE== 28
IMPURE_SIZE== 316

.PSECT \$CODE\$,NOWRT,2

013C 8F 00 6E 0000' CF 0000'
 00 2C 00002
 CF 00009
 01 D0 0000C
 04 00011

.ENTRY INIT COMMON, Save R2,R3,R4,R5 : 0450
MOVCS #0, (SP), #0, #316, USER_STATUS : 0488
MOVL #1, USER_STATUS : 0489
RET : 0491

; Routine Size: 18 bytes, Routine Base: \$CODE\$ + 0000

```
181 0492 1 GLOBAL ROUTINE SAVE_CONTEXT : NOVALUE =
182 0493 1
183 0494 1 !++
184 0495 1
185 0496 1 FUNCTIONAL DESCRIPTION:
186 0497 1
187 0498 1 This routine saves the reentrant context area in the context save
188 0499 1 area and initializes the context for a secondary operation.
189 0500 1
190 0501 1
191 0502 1 CALLING SEQUENCE:
192 0503 1 SAVE_CONTEXT ()
193 0504 1
194 0505 1 INPUT PARAMETERS:
195 0506 1 NONE
196 0507 1
197 0508 1 IMPLICIT INPUTS:
198 0509 1 ACP impure area
199 0510 1
200 0511 1 OUTPUT PARAMETERS:
201 0512 1 NONE
202 0513 1
203 0514 1 IMPLICIT OUTPUTS:
204 0515 1 NONE
205 0516 1
206 0517 1 ROUTINE VALUE:
207 0518 1 NONE
208 0519 1
209 0520 1 SIDE EFFECTS:
210 0521 1 NONE
211 0522 1
212 0523 1 --
213 0524 1
214 0525 2 BEGIN
215 0526 2
216 0527 2 MAP
217 0528 2 CONTEXT_SAVE : BITVECTOR; ! map saved copy of cleanup flags
218 0529 2
219 0530 2
220 0531 2 ! Check for excessive recursion in the ACP; then save the context and do the
221 0532 2 ! setup.
222 0533 2 !
223 0534 2 !
224 0535 2 IF .CONTEXT_SAVE NEQ 0
225 0536 2 THEN BUG_CHECK (ACPRECURS, FATAL, 'Attempted recursion in ACP secondary operation');
226 0537 2
227 0538 2 CH$MOVE (CONTEXT_SIZE, CONTEXT_START, CONTEXT_SAVE);
228 0539 2 CH$FILL (0, CONTEXT_SIZE, CONTEXT_START);
229 0540 2 CH$FILL (0, FIB$C_LENGTH, SECOND_FIB);
230 0541 2 CURRENT_FIB = SECOND_FIB;
231 0542 2 CONTEXT_SAVE[CLF_CLEANUP] = 1;
232 0543 2
233 0544 1 END; ! end of routine SAVE_CONTEXT
```

```
.EXTRN BUG$_ACPRECURS
```

				007C 00000	.ENTRY	SAVE CONTEXT, Save R2,R3,R4,R5,R6	: 0492			
		56	0000'	CF 9E 00002	MOVAB	CONTEXT_SAVE, R6				
				66 D5 00007	TSTL	CONTEXT_SAVE	: 0535			
				04 13 00009	BEQL	1\$				
				FEFF 0000B	BUGW		: 0536			
				0000* 0000D	.WORD	<BUG\$ ACPRECURS!4>				
		66	E4	A6	1C 28 0000F	1\$: MOV C3	#28, CONTEXT_START, CONTEXT_SAVE	: 0538		
		00		6E	00 2C 00014	MOV C5	#0, (SP), #0, #28, CONTEXT_START	: 0539		
				E4	A6	00019				
0040	8F	00		6E	00 2C 0001B	MOV C5	#0, (SP), #0, #64, SECOND_FIB	: 0540		
				FF74	C6	00022				
				FC	A6	FF74	C6 9E 00025	MOVAB	SECOND_FIB, CURRENT_FIB	: 0541
				01	A6	02	88 0002B	BISB2	#2, CONTEXT_SAVE+1	: 0542
						04	0002F	RET	: 0544	

: Routine Size: 48 bytes, Routine Base: \$CODE\$ + 0012

```

235 0545 1 GLOBAL ROUTINE RESTORE_CONTEXT : NOVALUE =
236 0546 1
237 0547 1 !**
238 0548 1
239 0549 1 FUNCTIONAL DESCRIPTION:
240 0550 1
241 0551 1     This routine restores the reentrant context area from the context save
242 0552 1     area.
243 0553 1
244 0554 1
245 0555 1 CALLING SEQUENCE:
246 0556 1     RESTORE_CONTEXT ()
247 0557 1
248 0558 1 INPUT PARAMETERS:
249 0559 1     NONE
250 0560 1
251 0561 1 IMPLICIT INPUTS:
252 0562 1     ACP impure area
253 0563 1
254 0564 1 OUTPUT PARAMETERS:
255 0565 1     NONE
256 0566 1
257 0567 1 IMPLICIT OUTPUTS:
258 0568 1     NONE
259 0569 1
260 0570 1 ROUTINE VALUE:
261 0571 1     NONE
262 0572 1
263 0573 1 SIDE EFFECTS:
264 0574 1     NONE
265 0575 1
266 0576 1 --
267 0577 1
268 0578 2 BEGIN
269 0579 2
270 0580 2 MAP
271 0581 2     CONTEXT_SAVE      : BITVECTOR;      ! map saved copy of cleanup flags
272 0582 2
273 0583 2
274 0584 2 ! Check for excessive unstacking in the ACP; then restore the context.
275 0585 2 !
276 0586 2
277 0587 2 IF .CONTEXT_SAVE EQL 0
278 0588 2 THEN BUG_CHECK (ACPUNSTAK, FATAL, 'Attempted unstack in ACP primary context');
279 0589 2
280 0590 2 CHSMOVE (CONTEXT_SIZE, CONTEXT_SAVE, CONTEXT_START);
281 0591 2 CLEANUP_FLAGS[CLF_CLEANUP] = 0;
282 0592 2 CONTEXT_SAVE = 0;
283 0593 2
284 0594 1 END;                                     ! end of routine RESTORE_CONTEXT

```

```

                                .EXTRN  BUG$_ACPUNSTAK
                                .ENTRY  RESTORE_CONTEXT, Save R2,R3,R4,R5,R6      ; 0545
56      0000'  CF 007C 0000      MOVAB  CONTEXT_SAVE, R6
                                9E 0002

```

```

        66 D5 00007      TSTL   CONTEXT_SAVE      : 0587
        04 12 00009      BNEQ   1$
        FEFF 0000B      BUGW
        0000* 0000D      .WORD   <BUG$ ACPUNSTAK!4>
E4  A6          E5  66  1C  28 0000F  1$:  MOV C3  #28, CONTEXT_SAVE, CONTEXT_START : 0590
        02 8A 00014      BIC B2  #2, CLEANUP_FLAGS+1      : 0591
        66 D4 00018      CLRL   CONTEXT_SAVE      : 0592
        04 0001A      RET      : 0594
    
```

: Routine Size: 27 bytes, Routine Base: \$CODE\$ + 0042

```

: 285      0595 1
: 286      0596 1 END
: 287      0597 0 ELUDOM
    
```

PSECT SUMMARY

Name	Bytes	Attributes
\$LOCKEDD1\$	328	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	93	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
. ABS	0	NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0)

Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	4	0	1000	00:01.9

COMMAND QUALIFIERS

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

```

: Size:          93 code + 328 data bytes
: Run Time:      00:07.3
: Elapsed Time:  00:24.4
: Lines/CPU Min: 4940
: Lexemes/CPU-Min: 12223
: Memory Used:  73 pages
: Compilation Complete
    
```

					CHKSUM LIS				
		ACPCNTR LIS			CHKPRO LIS				
	FCPOEF B32								DEACCS LIS
				BADSCN LIS					
					CLEUP LIS		CPYNAM LIS		
					CHKHOR LIS		COMMON LIS		CREHOR LIS
									CREWIN LIS
			ALLOB LIS						
	ACCESS LIS			CHKDMD LIS					
									DELETE LIS
							CREATE LIS	CREFCB LIS	