



```

CCCCCCCC RRRRRRRR EEEEEEEEEE AAAAAA TTTTTTTTTT EEEEEEEEEE
CCCCCCCC RRRRRRRR EEEEEEEEEE AAAAAA TTTTTTTTTT EEEEEEEEEE
CC        RR      RR EE          AA      AA TTT      TT EE
CC        RR      RR EE          AA      AA TTT      TT EE
CC        RR      RR EE          AA      AA TTT      TT EE
CC        RR      RR EE          AA      AA TTT      TT EE
CC        RRRRRRRR EEEEEEEE AA      AA TTT      TT EEEEEEEE
CC        RRRRRRRR EEEEEEEE AA      AA TTT      TT EEEEEEEE
CC        RR  RR   EE          AAAAAAAAAA TTT      TT EE
CC        RR  RR   EE          AAAAAAAAAA TTT      TT EE
CC        RR      RR EE          AA      AA TTT      TT EE
CC        RR      RR EE          AA      AA TTT      TT EE
CCCCCCCC RR      RR EEEEEEEEEE AA      AA TTT      TT EEEEEEEEEE
CCCCCCCC RR      RR EEEEEEEEEE AA      AA TTT      TT EEEEEEEEEE

```

```

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 III!II SSSSSSSS

```

```

1 0001 0 MODULE CREATE (
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 module processes the create function. It creates a file with the
38 0038 1 attributes requested, enters it in a directory if desired, and
39 0039 1 accesses it if requested.
40 0040 1
41 0041 1 ENVIRONMENT:
42 0042 1
43 0043 1 STARLET operating system, including privileged system services
44 0044 1 and internal exec routines.
45 0045 1
46 0046 1 --
47 0047 1
48 0048 1
49 0049 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 28-Mar-1977 15:05
50 0050 1
51 0051 1 MODIFIED BY:
52 0052 1
53 0053 1 V03-005 LMP0241 L. Mark Pilant, 26-Apr-1984 10:42
54 0054 1 Include the FIB in the MAKE_NAMEBLOCK routine call.
55 0055 1
56 0056 1 V03-004 LMP0154 L. Mark Pilant, 13-Sep-1983 13:17
57 0057 1 Set up a default protection in the created file header

```

```
58      0058 1      from the process default protection.
59      0059 1
60      0060 1      V03-003 ACG0329      Andrew C. Goldstein,      12-Apr-1983  16:05
61      0061 1      Fold long UIC's into [377,377]
62      0062 1
63      0063 1      V03-002 ACG53759      Andrew C. Goldstein,      18-Feb-1983  16:43
64      0064 1      Update revision date and count on ENTER operation
65      0065 1
66      0066 1      V03-001 LMP0018      L. Mark Pilant,          31-Mar-1982  13:10
67      0067 1      Modify to use a local copy of the window complete flag.
68      0068 1
69      0069 1      V02-004 LMP0005      L. Mark Pilant,          29-Dec-1981  14:30
70      0070 1      Added byte limit quota check on window creation, also a
71      0071 1      remap is done if the create did an initial allocation and
72      0072 1      Cathedral windows are desired.
73      0073 1
74      0074 1      V02-003 ACG0247      Andrew C. Goldstein,      23-Dec-1981  1:11
75      0075 1      Set revision date to creation date
76      0076 1
77      0077 1      V02-002 ACG0167      Andrew C. Goldstein,      7-May-1980   18:48
78      0078 1      Previous revision history moved to F11A.REV
79      0079 1      **
80      0080 1
81      0081 1
82      0082 1      LIBRARY 'SYSS$LIBRARY:LIB.L32';
83      0083 1      REQUIRE 'SRCS:F11A.SRC';
```

```

85 0398 1 GLOBAL ROUTINE CREATE =
86 0399 1
87 0400 1 **
88 0401 1
89 0402 1 FUNCTIONAL DESCRIPTION:
90 0403 1
91 0404 1 This routine processes the CREATE function. It creates a file with the
92 0405 1 attributes requested, enters it in a directory if desired, and
93 0406 1 accesses the file if requested.
94 0407 1
95 0408 1 CALLING SEQUENCE:
96 0409 1 CREATE ( )
97 0410 1
98 0411 1 INPUT PARAMETERS:
99 0412 1 NONE
100 0413 1
101 0414 1 IMPLICIT INPUTS:
102 0415 1 CURRENT_VCB: VCB of volume
103 0416 1 IO_PACKET: packet of this I/O request
104 0417 1
105 0418 1 OUTPUT PARAMETERS:
106 0419 1 NONE
107 0420 1
108 0421 1 IMPLICIT OUTPUTS:
109 0422 1 PRIMARY_FCB: FCB of file if accessed
110 0423 1 CURRENT_WINDOW: window of file if accessed
111 0424 1 USER_STATUS: I/O status block of user
112 0425 1
113 0426 1 ROUTINE VALUE:
114 0427 1 1 if successful
115 0428 1 0 if error
116 0429 1
117 0430 1 SIDE EFFECTS:
118 0431 1 File created, blocks allocated, directory modified, file accessed, etc.
119 0432 1
120 0433 1 --
121 0434 1
122 0435 2 BEGIN
123 0436 2
124 0437 2 LOCAL
125 0438 2 PACKET : REF BBLOCK, ! address of I/O packet
126 0439 2 ABD : REF BBLOCKVECTOR [ ,ABD$C_LENGTH],
127 0440 2 : buffer descriptors
128 0441 2 FIB : REF BBLOCK, ! file identification block
129 0442 2 RESULT_LENGTH, : length of result string from ENTER
130 0443 2 RESULT : VECTOR [20, BYTE], ! result string from ENTER
131 0444 2 NAMEBLOCK : BBLOCK [NMB$C_LENGTH], ! name block to build RAD-50 name
132 0445 2 IDENT_AREA : REF BBLOCK, ! pointer to file header ident area
133 0446 2 PCB : REF BBLOCK, ! Process Control Block address
134 0447 2 ARB : REF BBLOCK, ! access rights block of caller
135 0448 2 MAP_AREA : REF BBLOCK, ! file header map area
136 0449 2 IDX_FCB : REF BBLOCK, ! FCB of index file
137 0450 2 FCB : REF BBLOCK, ! FCB address
138 0451 2 HEADER : REF BBLOCK, ! address of file header
139 0452 2 FUNCTION : BLOCK [1]; ! function code qualifiers
140 0453 2
141 0454 2 EXTERNAL

```

```

142 0455 2 USER STATUS : VECTOR, : I/O status block of user
143 0456 2 CURRENT_VCB : REF BBLOCK, : VCB of volume
144 0457 2 PRIMARY_FCB : REF BBLOCK, : FCB of file
145 0458 2 CURRENT_WINDOW : REF BBLOCK, : window for file
146 0459 2 IO_PACKET : REF BBLOCK, : I/O request packet
147 0460 2 FILE_HEADER : REF BBLOCK, : global address of file header
148 0461 2 NEW_FID, : : unrecorded file ID
149 0462 2 HEADER_LBN, : : LBN of current file header
150 0463 2 SUPER_FID : BBLOCK, : file ID of superseded file
151 0464 2 SECOND_FIB : BBLOCK, : FIB for secondary operation
152 0465 2 CLEANUP_FLAGS : BITVECTOR, : cleanup action flags
153 0466 2 SCH$GL_PCBVEC : REF VECTOR ADDRESSING_MODE (ABSOLUTE); : PCB vector
154 0467 2
155 0468 2 EXTERNAL ROUTINE
156 0469 2 GET_FIB, : get FIB for operation
157 0470 2 CHECK_PROTECT, : check file protection
158 0471 2 CREATE_HEADER, : create a file ID and header
159 0472 2 CHECKSUM, : compute header checksum
160 0473 2 MARK_DIRTY, : mark buffer for write-back
161 0474 2 WRITE_HEADER, : write file header
162 0475 2 READ_HEADER, : read file header
163 0476 2 ENTER, : enter file in directory
164 0477 2 COPY_NAME, : copy file name to result string
165 0478 2 MAKE_NAMEBLOCK, : convert name string into RAD-50 name block
166 0479 2 GET_TIME, : get system date and time string
167 0480 2 CREATE_FCB, : create an FCB
168 0481 2 CREATE_WINDOW, : create a window
169 0482 2 MAKE_ACCESS, : complete the access
170 0483 2 MARKDEL_FCB, : mark FCB for delete
171 0484 2 WRITE_ATTRIB, : write attributes
172 0485 2 EXTEND, : extend the file
173 0486 2 SAVE_CONTEXT, : save reentrant context area
174 0487 2 RESTORE_CONTEXT, : restore reentrant context area
175 0488 2 MARK_DELETE, : mark file for delete
176 0489 2 NEXT_HEADER, : read next extension file header
177 0490 2 UPDATE_FCB, : update contents of FCB
178 0491 2 REMAP_FILE; : remap the file completely
179 0492 2
180 0493 2
181 0494 2 ! Enable the deaccess cleanup if an access is taking place.
182 0495 2 !
183 0496 2
184 0497 2 PACKET = .IO_PACKET;
185 0498 2 FUNCTION = .PACKET[IRP$W_FUNC];
186 0499 2 IF .FUNCTION[IOSV_ACCESS]
187 0500 2 THEN
188 0501 2 BEGIN
189 0502 2 CLEANUP_FLAGS[CLF_ZCHANNEL] = 1;
190 0503 2 CLEANUP_FLAGS[CLF_DELWINDOW] = 1;
191 0504 2 END;
192 0505 2
193 0506 2 ! Set up pointers to interesting control blocks.
194 0507 2 !
195 0508 2
196 0509 2 PCB = .SCH$GL_PCBVEC[(IO_PACKET[IRP$L_PID])<0,16>];
197 0510 2 ABD = .BBLOCK[.PACKET[IRP$L_SVAPTE], XIB$L_DESCRIPTOR];
198 0511 2 ! pointer to buffer descriptors

```

: R

```
199 0512 2 FIB = GET_FIB (.ABD); . pointer to FIB
200 0513 2
201 0514 2 IF .FIB[FIB$V_TRUNC]
202 0515 3 OR (.FUNCTION[IOSV_DELETE] AND NOT .FUNCTION[IOSV_ACCESS])
203 0516 3 OR (NOT .FUNCTION[IOSV_CREATE]
204 0517 4 AND (.FIB[FIB$V_EXTEND]
205 0518 4 OR .PACKET[IRP$W_BCNT] GTR ABD$C_ATTRIB
206 0519 4 OR .FUNCTION[IOSV_ACCESS]
207 0520 4 )
208 0521 3 )
209 0522 2 THEN ERR_EXIT (SS$BADPARAM);
210 0523 2
211 0524 2 IF .CURRENT_VCB[VCB$V_NOALLOC]
212 0525 2 THEN ERR_EXIT (SS$WRITLCK);
213 0526 2
214 0527 2 ! Do the create if requested. Start by allocating a file number from the
215 0528 2 ! index file bitmap and reading in the initial file header.
216 0529 2 !
217 0530 2
218 0531 2 IF .FUNCTION[IOSV_CREATE]
219 0532 2 THEN
220 0533 2 BEGIN
221 0534 2 CHECK_PROTECT (CREATE_ACCESS, 0, 0);
222 0535 2
223 0536 2 HEADER = CREATE_HEADER ();
224 0537 2 FIB[FIB$W_FID_NUM] = .HEADER[FH1$W_FID_NUM];
225 0538 2 FIB[FIB$W_FID_SEQ] = .HEADER[FH1$W_FID_SEQ];
226 0539 2 FIB[FIB$W_FID_RVN] = 0;
227 0540 2
228 0541 2 ! Now build an initialized file header in the buffer.
229 0542 2 !
230 0543 2
231 0544 2 HEADER[FH1$B_IDOFFSET] = FH1$C_LENGTH / 2;
232 0545 2 HEADER[FH1$B_MPOFFSET] = (FH1$C_LENGTH + FH1$C_LENGTH) / 2;
233 0546 2 HEADER[FH1$W_STRUCLEV] = FH1$C_LEVEL;
234 0547 2 ARB = .PACKET[IRP$L_ARB];
235 0548 2 HEADER[FH1$B_UICMEMBER] = .(ARB[ARB$L_UIC]) < 0, 8 >;
236 0549 2 HEADER[FH1$B_UICGROUP] = .(ARB[ARB$L_OIC]) < 16, 8 >;
237 0550 2 IF .(ARB[ARB$L_UIC]) < 8, 8 > NEQ 0
238 0551 2 OR .(ARB[ARB$L_UIC]) < 24, 8 > NEQ 0
239 0552 2 THEN
240 0553 2 BEGIN
241 0554 2 HEADER[FH1$B_UICMEMBER] = -1;
242 0555 2 HEADER[FH1$B_UICGROUP] = -1;
243 0556 2 END;
244 0557 2 HEADER[FH1$W_FILEPROT] = .PCB[PCB$L_DEFPROT];
245 0558 2
246 0559 2 CH$FILL (0, 512 - $BYTEOFFSET(FH1$W_FILECHAR), HEADER[FH1$W_FILECHAR]);
247 0560 2
248 0561 2 IF .FUNCTION[IOSV_DELETE]
249 0562 2 THEN HEADER[FH1$V_MARKDEL] = 1;
250 0563 2
251 0564 2 IF .CLEANUP_FLAGS[CLF_SPOOLFILE]
252 0565 2 THEN HEADER[FH1$V_SPOOL] = 1;
253 0566 2
254 0567 2 MAP_AREA = .HEADER + FH1$C_LENGTH + FH1$C_LENGTH;
255 0568 2 MAP_AREA[FH1$B_COUNTSIZE] = 1;
```

```
256 0569 3 MAP_AREA[FM1$B_LBNSIZE] = 3;
257 0570 3 MAP_AREA[FM1$B_INUSE] = 0;
258 0571 3 MAP_AREA[FM1$B_AVAIL] = (512 - FH1$C_LENGTH - FI1$C_LENGTH - FM1$C_LENGTH - 2) / 2;
259 0572 3
260 0573 3 NEW_FID = 0; ! new file ID is no longer unrecorded
261 0574 3 MARR DIRTY (.HEADER);
262 0575 3 CLEARUP_FLAGS[CLF_DELFIL] = 1;
263 0576 3 FILE_HEADER = .HEADER; ! record header address for cleanup
264 0577 3 CHECKSUM (.HEADER);
265 0578 3 END;
266 0579 3
267 0580 3 ! If a non-zero directory ID is supplied, enter the file in the directory.
268 0581 3 ! Otherwise, just copy down the name string (if any) into the result string.
269 0582 3 ! Note that the directory operation is nooped on spool file operations.
270 0583 3 ! Then convert the result string, from whatever source, into RAD-50 into the
271 0584 3 ! file header ident area.
272 0585 3
273 0586 3
274 0587 3 IF .FIB[FIB$W_DID_NUM] NEQ 0 AND NOT .CLEANUP_FLAGS[CLF_SPOOLFILE]
275 0588 3 THEN
276 0589 3 ENTER (.ABD, .FIB, RESULT_LENGTH, RESULT)
277 0590 3 ELSE
278 0591 3 BEGIN
279 0592 3 KERNEL_CALL (COPY_NAME, .ABD);
280 0593 3 RESULT_LENGTH = .ABD[ABD$C_NAME, ABD$W_COUNT];
281 0594 3 CHSMOVE (.RESULT_LENGTH,
282 0595 3 .ABD[ABD$C_NAME, ABD$W_TEXT] + ABD[ABD$C_NAME, ABD$W_TEXT] + 1, RESULT);
283 0596 3 END;
284 0597 3
285 0598 3 ! Read the file header. Then copy in the generated file name string,
286 0599 3 ! and update the revision date and time and revision count.
287 0600 3
288 0601 3
289 0602 3 HEADER = READ HEADER (FIB[FIB$W_FID], 0);
290 0603 3 MAKE NAMEBLOCK (.FIB, .RESULT_LENGTH, RESULT, NAMEBLOCK);
291 0604 3 IDENT_AREA = .HEADER + .HEADER[FH1$B_IDOFFSET]*2;
292 0605 3 CHSMOVE (10, NAMEBLOCK[NMBSW_NAME], IDENT_AREA[F11$W_FILENAME]);
293 0606 3 GET TIME (IDENT_AREA[F11$T_REVDATE]);
294 0607 3 IDENT_AREA[F11$W_REVISION] = .IDENT_AREA[F11$W_REVISION] + 1;
295 0608 3
296 0609 3 ! If a new file is being created, zero the revision count again, and
297 0610 3 ! copy the date into the creation date. Then write the attributes.
298 0611 3
299 0612 3
300 0613 3 IF .FUNCTION[IOSV_CREATE]
301 0614 3 THEN
302 0615 3 BEGIN
303 0616 3 IDENT_AREA[F11$W_REVISION] = 0;
304 0617 3 CHSMOVE (F11$S_CREDATE+F11$S_CREATIME, IDENT_AREA[F11$T_REVDATE], IDENT_AREA[F11$T_CREDATE]);
305 0618 3
306 0619 3 IF .PACKET[IRPSW_BCNT] GTR ABD$C_ATTRIB
307 0620 3 THEN WRITE_ATTRIB (.HEADER, .ABD, 0);
308 0621 3
309 0622 3 ! If access is requested, access the file.
310 0623 3
311 0624 3
312 0625 3 IF .FUNCTION[IOSV_ACCESS]
```



```

313      0626 3      THEN
314      0627 4          BEGIN
315      0628 4          FCB = KERNEL_CALL (CREATE_FCB, .HEADER);
316      0629 4          PRIMARY_FCB = .FCB;
317      0630 4
318      P 0631 4          CURRENT_WINDOW = KERNEL_CALL (CREATE_WINDOW, .FIB[FIB$L ACCTL],
319      0632 4              .FIB[FIB$B WSIZE], .HEADER, .PACKET[IRP$L PID], .FCB);
320      0633 4          IF .CURRENT_WINDOW EQL 0 THEN ERR_EXIT (SS$ EXBYTLM);
321      0634 4          KERNEL_CALL (MAKE_ACCESS, .FCB, .CURRENT_WINDOW, .ABD);
322      0635 4          IF .FUNCTION[IOSV_DELETE]
323      0636 4              THEN KERNEL_CALL (MARKDEL_FCB, .FCB);
324      0637 4          END;
325      0638 3
326      0639 3      ! Now extend the file if requested.
327      0640 3      !
328      0641 3
329      0642 3          IF .FIB[FIB$V EXTEND] THEN EXTEND (.FIB, .HEADER);
330      0643 3          HEADER = .FILE_HEADER;
331      0644 3          KERNEL_CALL (UPDATE_FCB, .HEADER);
332      0645 3          END;
333      0646 2
334      0647 2      CHECKSUM (.HEADER);
335      0648 2      WRITE_HEADER ();
336      0649 2
337      0650 2      ! Perform the remap operation if necessary to account for any initial extend.
338      0651 2      !
339      0652 2
340      0653 2      IF .FUNCTION[IOSV_ACCESS] AND .FIB[FIB$V_EXTEND]
341      0654 2      THEN IF .CURRENT_WINDOW[WCBSV_CATHEDRAL]
342      0655 2      THEN REMAP_FILE (?);
343      0656 2
344      0657 2      ! If this is a supersede operation, delete the file that was removed during
345      0658 2      ! the enter operation above. This must be done last since we cannot undo
346      0659 2      ! a delete in cleaning up from a subsequent error. We first copy the primary
347      0660 2      ! context into the context save area since this is a secondary operation.
348      0661 2      !
349      0662 2
350      0663 2      IF TESTBITSC (CLEANUP_FLAGS[CLF_SUPERSEDE])
351      0664 2      THEN
352      0665 3          BEGIN
353      0666 3              USER_STATUS[0] = SS$_SUPERSEDE;
354      0667 3              SAVE_CONTEXT ();
355      0668 3              CH$COPY (FID$C_LENGTH, SUPER_FID, 0,
356      0669 3                  FIB$C_LENGTH - $BYTEOFFSET (FIB$W_FID), SECOND_FIB[FIB$W_FID]);
357      0670 3              MARK_DELETE (SECOND_FIB);
358      0671 3              RESTORE_CONTEXT ();
359      0672 3          END;
360      0673 2
361      0674 2      RETURN 1;
362      0675 2
363      0676 2
364      0677 1      END;

```

! end of routine CREATE

.TITLE CREATE  
.IDENT \V04-000\

```

        .EXTRN USER STATUS, CURRENT_VCB
        .EXTRN PRIMARY_FCB, CURRENT_WINDOW
        .EXTRN IO_PACKET, FILE_HEADER
        .EXTRN NEW_FID, HEADER_LBN
        .EXTRN SUPER_FID, SECOND_FIB
        .EXTRN CLEANUP_FLAGS, SCH$GL_PCBVEC
        .EXTRN GET_FIB, CHECK_PROTECT
        .EXTRN CREATE_HEADER, CHECKSUM
        .EXTRN MARK_DIRTY, WRITE_HEADER
        .EXTRN READ_HEADER, ENTER
        .EXTRN COPY_NAME, MAKE_NAMEBLOCK
        .EXTRN GET_TIME, CREATE_FCB
        .EXTRN CREATE_WINDOW, MAKE_ACCESS
        .EXTRN MARKDEC_FCB, WRITE_ATTRIB
        .EXTRN EXTEND, SAVE_CONTEXT
        .EXTRN RESTORE_CONTEXT
        .EXTRN MARK_DELETE, NEXT_HEADER
        .EXTRN UPDATE_FCB, REMAP_FILE
        .EXTRN SYSSCMRNL

        .PSECT $CODE$,NOWRT,2

        OFFC 00000
        .ENTRY CREATE, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-
            R11
            MOVAB -64(SP), SP
            MOVL IO_PACKET, R0
            MOVL R0, PACKET
            MOVZWL 32(PACKET), FUNCTION
            BBC #6, FUNCTION, 1$
            BISW2 #1026, CLEANUP_FLAGS+2
            MOVL @#SCH$GL_PCBVEC, R1
            MOVZWL 12(R0), R0
            MOVL (R1)[R0], PCB
            MOVL @44(PACKET), ABD
            PUSHL ABD
            CALLS #1, GET_FIB
            MOVL R0, FIB
            BLBS 23(FIB), 3$
            BBC #8, FUNCTION, 2$
            BBC #6, FUNCTION, 3$
            TSTB FUNCTION
            BLSS 4$
            TSTB 22(FIB)
            BLSS 3$
            CMPW 50(PACKET), #5
            BGTRU 3$
            BBC #6, FUNCTION, 4$
            CHMU #20
            RET
            MOVL CURRENT_VCB, R0
            BBC #4, 11(R0), 5$
            CHMU #604
            RET
            TSTB FUNCTION
            BLSS 6$
            BRW 11$
            CLRQ -(SP)
            : 0534

07      0000G  SE      CO      AE      9E      00002
        50      0000G  CF      D0      00006
        59      50      D0      0000B
        5B      20      A9      3C      0000E
        5B      06      E1      00012
        CF      0402  8F      AB      00016
        51      00000000G  9F      D0      0001D 1$:
        50      0C      A0      3C      00024
        52      6140  D0      00028
        5A      2C      B9      D0      0002C
        5A      DD      00030
        0C00G  CF      01      FB      00032
        58      50      D0      00037
        1B      17      A8      E8      0003A
        04      5B      08      E1      0003E
        13      5B      06      E1      00042
        5B      95      00046 2$:
        12      19      00048
        16      A8      95      0004A
        0A      19      0004D
        05      32      A9      B1      0004F
        04      1A      00053
        03      5B      06      E1      00055
        14      BF      00059 3$:
        04      0005B
        05      0B      50      0000G  CF      D0      0005C 4$:
        A0      04      E1      00061
        025C  8F      BF      00066
        04      0006A
        5B      95      0006B 5$:
        03      19      0006D
        008C  31      0006F
        7E      7C      00072 6$:

```

				03	DD	00074		PUSHL	#3		
	0000G	CF		03	FB	00076		CALLS	#3, CHECK PROTECT		
	0000G	CF		00	FB	0007B		CALLS	#0, CREATE_HEADER		0536
		57		50	DO	00080		MOVL	R0, HEADER		
	04	A8		A7	DO	00083		MOVL	2(HEADER), 4(FIB)		0537
			02	A8	B4	00088		CLRW	8(FIB)		0539
		67	2E17	8F	B0	0008B		MOVW	#11799, (HEADER)		0544
	06	A7	0101	8F	B0	00090		MOVW	#257, 6(HEADER)		0546
		50		A9	DO	00096		MOVL	88(PACKET), ARB		0547
	08	A7		A0	90	0009A		MOVW	56(ARB), 8(HEADER)		0548
	09	A7		A0	90	0009F		MOVW	58(ARB), 9(HEADER)		0549
				A0	95	000A4		TSTB	57(ARB)		0550
				05	12	000A7		BNEQ	7\$		
				A0	95	000A9		TSTB	59(ARB)		0551
	08	A7	FFFF	06	13	000AC		BEQL	8\$		
	0A	A7	0:14	8F	B0	000AE	7\$:	MOVW	#65535, 8(HEADER)		0554
01F4	8F			C2	B0	000B4	8\$:	MOVW	276(PCB), 10(HEADER)		0557
		6E		00	2C	000BA		MOVCS	#0, (SP), #0, #500, 12(HEADER)		0559
				A7		000C1					
	05			08	E1	000C3		BBC	#8, FUNCTION, 9\$		0561
	0D	A7		8F	88	000C7		BISB2	#128, 13(HEADER)		0562
				CF	95	000CC	9\$:	TSTB	CLEANUP_FLAGS		0564
				04	18	000D0		BGEQ	10\$		
	0D	A7		10	88	000D2		BISB2	#16, 13(HEADER)		0565
		50	5C	A7	9E	000D6	10\$:	MOVAB	92(R7), MAP_AREA		0567
	06	A0	CC000301	8F	DO	000DA		MOVL	#-872414463, 6(MAP_AREA)		0568
				CF	D4	000E2		CLRL	NEW FID		0573
				57	DD	000E6		PUSHL	HEADER		0574
	0000G	CF		01	FB	000E8		CALLS	#1, MARK DIRTY		
	0000G	CF		20	88	000ED		BISB2	#32, CLEANUP_FLAGS+2		0575
	0000G	CF		57	DO	000F2		MOVL	HEADER, FILE_HEADER		0576
				57	DD	000F7		PUSHL	HEADER		0577
	0000G	CF		01	FB	000F9		CALLS	#1, CHECKSUM		
				AB	B5	000FE	11\$:	TSTW	10(FIB)		0587
				17	13	00101		BEQL	12\$		
				CF	95	00103		TSTB	CLEANUP_FLAGS		
				11	19	00107		BLSS	12\$		
				AE	9F	00109		PUSHAB	RESULT		0589
				AE	9F	0010C		PUSHAB	RESULT_LENGTH		
				58	DD	0010F		PUSHL	FIB		
				5A	DD	00111		PUSHL	ABD		
	0000G	CF		04	FB	00113		CALLS	#4, ENTER		
				23	11	00118		BRB	13\$		
				5A	DD	0011A	12\$:	PUSHL	ABD		0592
				01	DD	0011C		PUSHL	#1		
				5E	DD	0011E		PUSHL	SP		
				CF	9F	00120		PUSHAB	COPY_NAME		
	00000000G	9F		04	FB	00124		CALLS	#4, #SYSSCMKRNL		
		6E	12	AA	3C	0012B		MOVZWL	18(ABD), RESULT_LENGTH		0593
		51	10	AA	9E	0012F		MOVAB	16(ABD), R1		0595
		50		61	3C	00133		MOVZWL	(R1), R0		
2C	AE	01	A140	6E	28	00136		MOVCS	RESULT_LENGTH, 1(R1)[R0], RESULT		0594
				7E	D4	0013D	13\$:	CLRL	-(SP)		0602
				A8	9F	0013F		PUSHAB	4(FIB)		
	0000G	CF		02	FB	00142		CALLS	#2, READ HEADER		
		57		50	DO	00147		MOVL	R0, HEADER		
				04	AE	9F	0014A	PUSHAB	NAMEBLOCK		0603

				30	AE	9F	0014D		PUSHAB	RESULT		
				08	AE	DD	00150		PUSHL	RESULT_LENGTH		
					58	DD	00153		PUSHL	FIB		
	000CG	CF			04	FB	00155		CALLS	#4, MAKE_NAMEBLOCK		
		50			67	9A	0015A		MOVZBL	(HEADER)_R0	0604	
		56		6740	3E	0015D		MOVAV	(HEADER)[R0], IDENT_AREA			
66	OA	AE		0A	28	00161		MOVCS	#10, NAMEBLOCK+6, (IDENT_AREA)	0605		
	0000G	CF		0C	A6	9F	00166		PUSHAB	12(IDENT_AREA)	0606	
					01	FB	00169		CALLS	#1, GET_TIME		
				0A	A6	B6	0016E		INCW	10(IDENT_AREA)	0607	
					5B	95	00171		TSTB	FUNCTION	0613	
					03	19	00173		BLSS	14\$		
					00AF	31	00175		BRW	19\$		
				0A	A6	B4	00178	14\$:	CLRW	10(IDENT_AREA)	0616	
19	A6	OC	A6		0D	28	0017B		MOVCS	#13, 12(IDENT_AREA), 25(IDENT_AREA)	0617	
			05		32	A9	B1	00181	CMPL	50(PACKET), #5	0619	
						0B	1B	00185	BLEQU	15\$		
						7E	D4	00187	CLRL	-(SP)	0620	
				0480	8F	BB	00189		PUSHR	#*M<R7,R10>		
					03	FB	0018D		CALLS	#3, WRITE_ATTRIB		
6D	0000G	CF			06	E1	00192	15\$:	BBC	#6, FUNCTION, 17\$	0625	
		5B				57	DD	00196	PUSHL	HEADER	0628	
						01	DD	00198	PUSHL	#1		
						5E	DD	0019A	PUSHL	SP		
				0000G	CF	9F	0019C		PUSHAB	CREATE_FCB		
					04	FB	001A0		CALLS	#4, @#SYSS\$CMKRNL		
	00000000G	9F			50	DD	001A7		MOVL	R0, FCB		
		52			52	DD	001AA		MOVL	FCB, PRIMARY_FCB	0629	
	0000G	CF			52	DD	001AF		PUSHL	FCB	0632	
					0C	A9	DD	001B1	PUSHL	12(PACKET)		
						57	DD	001B4	PUSHL	HEADER		
						7E	03	A8	98	001B6		
						68	DD	001BA	PUSHL	3(FIB), -(SP)		
						05	DD	001BC	PUSHL	(FIB)		
						5E	DD	001BE	PUSHL	#5		
				0000G	CF	9F	001C0		PUSHL	SP		
					08	FB	001C4		PUSHAB	CREATE_WINDOW		
	00000000G	9F			50	DD	001CB		CALLS	#8, @#SYSS\$CMKRNL		
	0000G	CF			05	12	001D0		MOVL	R0, CURRENT_WINDOW		
						8F	BF	001D2	BNEQ	16\$	0633	
				2A14		04	001D6		CHMU	#10772		
						5A	DD	001D7	RET			
						CF	DD	001D9	PUSHL	ABD	0634	
				0000G		52	DD	001DD	PUSHL	CURRENT_WINDOW		
						03	DD	001DF	PUSHL	FCB		
						5E	DD	001E1	PUSHL	#3		
				0000G	CF	9F	001E3		PUSHL	SP		
					06	FB	001E7		PUSHAB	MAKE_ACCESS		
11	00000000G	9F			08	E1	001EE		CALLS	#6, @#SYSS\$CMKRNL		
		5B			52	DD	001F2		BBC	#8, FUNCTION, 17\$	0635	
						01	DD	001F4	PUSHL	FCB	0636	
						5E	DD	001F6	PUSHL	#1		
				0000G	CF	9F	001F8		PUSHL	SP		
					04	FB	001FC		PUSHAB	MARKDEL_FCB		
	00000000G	9F				A8	95	00203	CALLS	#4, @#SYSS\$CMKRNL		
				16		09	18	00206	TSTB	22(FIB)	0642	
						57	DD	00208	BGEQ	18\$		
									PUSHL	HEADER		

		0000G	CF		58	DD	0020A		PUSHL	FIB		
			57	0000G	02	FB	0020C		CALLS	#2, EXTEND		0643
					57	DD	00211	18\$:	MOVL	FILE HEADER, HEADER		0644
					01	DD	00216		PUSHL	HEADER		
					5E	DD	00218		PUSHL	#1		
					CF	9F	0021A		PUSHL	SP		
		00000000G	9F	0000G	04	FB	0021C		PUSHAB	UPDATE_FCB		
					57	DD	00220		CALLS	#4, @#SYSS\$CMKRNL		
					01	FB	00227	19\$:	PUSHL	HEADER		0647
		0000G	CF		00	FB	00229		CALLS	#1, CHECKSUM		
14		0000G	CF		00	FB	0022E		CALLS	#0, WRITE_HEADER		0648
			5B		06	E1	00233		BBC	#6, FUNCTION, 20\$		0653
				16	A8	95	00237		TSTB	22(FIB)		
					0F	18	0023A		BGEQ	20\$		
			50	0000G	CF	DD	0023C		MOVL	CURRENT_WINDOW, R0		0654
05		0B	A0		06	E1	00241		BBC	#6, 11(R0), 20\$		
		0000G	CF		00	FB	00246		CALLS	#0, REMAP_FILE		0655
24		0000G	CF		05	E5	0024B	20\$:	BBCC	#5, CLEANUP_FLAGS, 21\$		0663
		0000G	CF	0631	8F	3C	00251		MOVZWL	#1585, USER_STATUS		0666
		0000G	CF		00	FB	00258		CALLS	#0, SAVE_CONTEXT		0667
3C		00	CF		06	2C	0025D		MOVCS	#6, SUPER_FID, #0, #60, SECOND_FIB+4		0669
				0000G	CF		00264					
		0000G	CF	0000G	CF	9F	00267		PUSHAB	SECOND_FIB		0670
		0000G	CF		01	FB	0026B		CALLS	#1, MARK_DELETE		
		0000G	CF		00	FB	00270		CALLS	#0, RESTORE_CONTEXT		0671
			50		01	DD	00275	21\$:	MOVL	#1, R0		0675
					04	00278			RET			0677

: Routine Size: 633 bytes, Routine Base: \$CODE\$ + 0000

```
: 365      0678  1
: 366      0679  1 END
: 367      0680  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	633	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	62	0	1000	00:01.9

CREATE  
V04-000

M 13  
16-Sep-1984 00:53:11  
14-Sep-1984 12:29:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[F11A.SRC]CREATE.B32;1 Page 12 (2)

CR  
VO

COMMAND QUALIFIERS

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

: Size: 633 code + 0 data bytes  
: Run Time: 00:16.3  
: Elapsed Time: 00:46.9  
: Lines/CPU Min: 2506  
: Lexemes/CPU-Min: 17561  
: Memory Used: 240 pages  
: Compilation Complete

.....

FCPOEF B32	ACPCNTR LIS	CHKSUM LIS	CHKPRO LIS	DEACCS LIS
BADSEN LIS	CLENUP LIS	CPYAM LIS	CHKHDR LIS	COMMON LIS
CREHDR LIS	CREWIN LIS	ACCESS LIS	ALLOB LIS	CHKDMD LIS
CREATE LIS	CREFCB LIS	DELETE LIS		