


```

LL      IIIIII  BBBB8888  UU      UU  NN      NN  LL      FFFFFFFF  IIIIII  LL
LL      IIIIII  BBBB8888  UU      UU  NN      NN  LL      FFFFFFFF  IIIIII  LL
LL      II      BB      BB  UU      UU  NN      NN  LL      FF      II      LL
LL      II      BB      BB  UU      UU  NN      NN  LL      FF      II      LL
LL      II      BB      BB  UU      UU  NNNN     NN  LL      FF      II      LL
LL      II      BB      BB  UU      UU  NNNN     NN  LL      FF      II      LL
LL      II      BB      BB  UU      UU  NN      NN  LL      FFFFFFFF  II      LL
LL      II      BB      BB  UU      UU  NN      NN  LL      FFFFFFFF  II      LL
LL      II      BB      BB  UU      UU  NN      NN  LL      FF      II      LL
LL      II      BB      BB  UU      UU  NN      NN  LL      FF      II      LL
LL      II      BB      BB  UU      UU  NN      NN  LL      FF      II      LL
LLLLLLLL  IIIIII  BBBB8888  UUUUUUUUUU  NN      NN  LLLLLLLLLL  FF      IIIIII  LLLLLLLLLL
LLLLLLLL  IIIIII  BBBB8888  UUUUUUUUUU  NN      NN  LLLLLLLLLL  FF      IIIIII  LLLLLLLLLL

```

```

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
LLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLL  IIIIII  SSSSSSSS

```

```
1 0001 0 %TITLE 'LIBSUNLOCK_FILE - Clear deaccess lock on file'  
2 0002 0 MODULE LIBSUNLOCK_FILE ( ! Clear deaccess lock on file  
3 0003 0 IDENT = 'V04-000' ! File: LIBUNLFIL.B32 Edit: 1-001  
4 0004 0 ) =  
5 0005 1 BEGIN  
6 0006 1  
7 0007 1 *****  
8 0008 1 *  
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
11 0011 1 * ALL RIGHTS RESERVED. *  
12 0012 1 * *  
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
18 0018 1 * TRANSFERRED. *  
19 0019 1 * *  
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
22 0022 1 * CORPORATION. *  
23 0023 1 * *  
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
26 0026 1 * *  
27 0027 1 *  
28 0028 1 *****  
29 0029 1  
30 0030 1  
31 0031 1 ++  
32 0032 1 FACILITY: General Utility Library  
33 0033 1  
34 0034 1 ABSTRACT:  
35 0035 1  
36 0036 1 This routine clears a deaccess lock on a file.  
37 0037 1  
38 0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant  
39 0039 1  
40 0040 1 AUTHOR: Martin L. Jack, CREATION DATE: 23-Dec-1981  
41 0041 1  
42 0042 1 MODIFIED BY:  
43 0043 1  
44 0044 1 1-001 - Original from LIBACP.B32. MLJ 23-Dec-1981  
45 0045 1 --  
46 0046 1
```

UT

Th
ME

```

48 0047 1 %SBTTL 'Declarations'
49 0048 1
50 0049 1 SWITCHES:
51 0050 1
52 0051 1
53 0052 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
54 0053 1
55 0054 1
56 0055 1 LINKAGES:
57 0056 1
58 0057 1 LINKAGE
59 0058 1 LINKAGE_JSB_2_2 = JSB(REGISTER=0;REGISTER=1,REGISTER=2);
60 0059 1
61 0060 1 TABLE OF CONTENTS:
62 0061 1
63 0062 1
64 0063 1 FORWARD ROUTINE
65 0064 1 LIB$UNLOCK_FILE; ! Clear deaccess lock on file
66 0065 1
67 0066 1
68 0067 1 INCLUDE FILES:
69 0068 1
70 0069 1
71 0070 1 LIBRARY 'SYS$LIBRARY:LIB'; ! System symbols
72 0071 1
73 0072 1 !*! REQUIRE 'RTLIN:RTLPSECT'; ! Define PSECT declarations macros
74 0073 1
75 0074 1
76 0075 1 MACROS:
77 0076 1
78 0077 1 NONE
79 0078 1
80 0079 1 EQUATED SYMBOLS:
81 0080 1
82 0081 1 NONE
83 0082 1
84 0083 1 FIELDS:
85 0084 1
86 0085 1 NONE
87 0086 1
88 0087 1 PSECTS:
89 0088 1
90 0089 1 !*! DECLARE_PSECTS (LIB); ! Declare PSECTS for LIB$ facility
91 0090 1 PSECT
92 0091 1 CODE = _LIB$CODE (READ, NOWRITE, EXECUTE, SHARE, PIC, ADDRESSING_MODE (WORD_RELATIVE)),
93 0092 1 PLIT = _LIB$CODE (READ, NOWRITE, EXECUTE, SHARE, PIC, ADDRESSING_MODE (WORD_RELATIVE)),
94 0093 1 OWN = _LIB$DATA (READ, WRITE, NOEXECUTE, NOSHARE, PIC, ADDRESSING_MODE (LONG_RELATIVE)),
95 0094 1 GLOBAL = _LIB$DATA (READ, WRITE, NOEXECUTE, NOSHARE, PIC, ADDRESSING_MODE (LONG_RELATIVE));
96 0095 1
97 0096 1 OWN STORAGE:
98 0097 1
99 0098 1 NONE
100 0099 1
101 0100 1 EXTERNAL REFERENCES:
102 0101 1
103 0102 1
104 0103 1 EXTERNAL ROUTINE

```

LIB\$UNLOCK_FILE LIB\$UNLOCK_FILE - Clear deaccess lock on file
V04-000 Declarations

N 8
16-Sep-1984 02:27:18
14-Sep-1984 13:34:30

VAX-11 Bliss-32 V4.0-742
[VM\$LIB.SRC]LIBUNLFIL.B32;1

```
: 105      0104 1      LIB$ANALYZE_SDESC_R2:      LINKAGE JSB_2_2,      ! Analyze descriptor
: 106      0105 1      LIB$FREE_EF;              ! Deallocate an event flag
: 107      0106 1      LIB$GET_EF;              ! Allocate an event flag
: 108      0107 1
: 109      0108 1      EXTERNAL LITERAL          ! Completion status codes
: 110      0109 1      LIB$INVARG;              ! Invalid argument
: 111      0110 1      LIB$INVFIL$PE;          ! Invalid file specification
```

```
0111 1 %SBTTL 'LIB$UNLOCK_FILE - Clear deaccess lock on file'
0112 1 GLOBAL ROUTINE LIB$UNLOCK_FILE (
0113 1     FILE_SPEC
0114 1 ) =
0115 1
0116 1
0117 1
0118 1
0119 1
0120 1
0121 1
0122 1
0123 1
0124 1
0125 1
0126 1
0127 1
0128 1
0129 1
0130 1
0131 1
0132 1
0133 1
0134 1
0135 1
0136 1
0137 1
0138 1
0139 1
0140 1
0141 1
0142 1
0143 1
0144 1
0145 1
0146 1
0147 1
0148 1
0149 1
0150 1
0151 1
0152 1
0153 1
0154 1
0155 1
0156 1
0157 1
0158 1
0159 1
0160 1
0161 1
0162 1
0163 1
0164 1
0165 1
0166 1
0167 1
```

++
FUNCTIONAL DESCRIPTION:
This routine clears a deaccess lock on a file.

CALLING SEQUENCE:
ret_status.wlc.v = LIB\$UNLOCK (file-spec.rt.dx)

FORMAL PARAMETERS:
FILE_SPEC Address of a descriptor for the file specification.
This string is a standard RMS file specification; it
must not contain a node name nor wild card characters;
it must reference a disk device. The string must be
no longer than 255 characters.

IMPLICIT INPUTS:
NONE

IMPLICIT OUTPUTS:
NONE

COMPLETION STATUS:
SS\$_WASSET Normal successful completion; file was unlocked
SS\$_WASCLR Normal successful completion; file was not locked
LIB\$_INVARG Required argument omitted, or file-spec longer than
255 characters
LIB\$_INVFILSPE File specification contained a node name or wildcard or
did not reference a disk device

LIB\$ANALYZE_SDESC errors
\$PARSE errors
\$SEARCH errors
\$ASSIGN errors
LIB\$GET_EF errors
\$QIO errors
\$DASSGN errors
LIB\$FREE_EF errors

SIDE EFFECTS:
Deaccess lock cleared on file.

--

```
170 0168 2 BEGIN
171 0169 2 LOCAL
172 0170 2 FAB: $FAB_DECL, ! FAB for $PARSE
173 0171 2 NAM: $NAM_DECL, ! NAM block for $PARSE
174 0172 2 ESA_BUFFER: VECTOR[NAM$C_MAXRSS, BYTE], ! Expanded string area
175 0173 2 TEMP_DESC: BLOCK[DSC$K_5_BLN, BYTE], ! Utility descriptor
176 0174 2 UCHAR: BLOCK[ATR$$_UCHAR, BYTE]
177 0175 2 VOLATILE, ! File characteristics
178 0176 2 FIB: BLOCK[FIB$C_LENGTH, BYTE], ! FIB
179 0177 2 FIB_DESC: VECTOR[2], ! Descriptor for FIB
180 0178 2 ATR: BLOCKVECTOR[2, 8, BYTE], ! Attribute descriptors
181 0179 2 IOSB: VECTOR[4, WORD], ! I/O status block
182 0180 2 CHANNEL: WORD, ! Channel number
183 0181 2 EFN, ! Event flag number
184 0182 2 STATUS_1, ! Status return
185 0183 2 STATUS_2, ! Status return
186 0184 2 STATUS_3, ! Status return
187 0185 2 STATUS_4, ! Status return
188 0186 2 STATUS_5, ! Status return
189 0187 2 STATUS_6, ! Status return
190 0188 2 STATUS_7, ! Status return
191 0189 2 FINAL_STATUS: ! Status return
192 0190 2 LABEL
193 0191 2 PROCESS; ! Block exited when processing complete
194 0192 2 BUILTIN
195 0193 2 ACTUALCOUNT, ! Return number of arguments
196 0194 2 TESTBITSC; ! Test if bit set, clear bit
197 0195 2
198 0196 2 !+
199 0197 2 ! Ensure that the required parameter is present.
200 0198 2 !-
201 0199 2
202 0200 2 IF ACTUALCOUNT() EQL 0 THEN RETURN LIB$_INVARG;
203 0201 2
204 0202 2 !+
205 0203 2 ! Initialize RMS structures required to do a $PARSE and $SEARCH.
206 0204 2 !-
207 0205 2
208 P 0206 2 $FAB_INIT(FAB=FAB,
209 0207 2 NAM=NAM);
210 P 0208 2 $NAM_INIT(NAM=NAM,
211 P 0209 2 ESA=ESA_BUFFER,
212 0210 2 ESS=NAM$C_MAXRSS);
213 0211 2
214 0212 2 !+
215 0213 2 ! Analyze the input descriptor and set up the FAB filename descriptor.
216 0214 2 !-
217 0215 2
218 0216 2 BEGIN ! block to use output registers
219 0217 2 REGISTER
220 0218 2 R1 = 1;
221 0219 2 R2 = 2;
222 0220 2
223 0221 2 STATUS_1 = LIB$ANALYZE_SDESC_R2(.FILE SPEC; R1, R2);
224 0222 2 IF NOT .STATUS_1 THEN RETURN .STATUS_T;
225 0223 2 IF .R1 GTRU 255 THEN RETURN LIB$_INVARG;
226 0224 2 FAB[FAB$B_FNS] = .R1;
```

```
227 0225 3 FAB[FAB$FNA] = .R2;
228 0226 2 END; ! block to use output registers
229 0227 2
230 0228 2 !+
231 0229 2 ! Parse the file specification to obtain the expanded name string.
232 0230 2 !-
233 0231 2
234 0232 2 STATUS_2 = $PARSE(FAB=FAB);
235 0233 2 IF NOT .STATUS_2 THEN RETURN .STATUS_2;
236 0234 2
237 0235 2 !+
238 0236 2 ! Perform various error checks on the file specification. It must not have a
239 0237 2 ! node name, must not contain wildcards, and must reference a disk device.
240 0238 2 !-
241 0239 2
242 0240 2 IF
243 0241 2 (.NAM[NAM$FNB] AND (NAM$M_WILDCARD OR NAM$M_NODE)) NEQ 0 OR
244 0242 2 NOT .BLOCK[FAB[FAB$DEV], DEV$V_RND; ,BYTE]
245 0243 2 THEN
246 0244 2 BEGIN
247 0245 2
248 0246 2 !+
249 0247 2 ! If the string contained a wildcard or a node name, internal RMS resources
250 0248 2 ! have been consumed by $PARSE. Execute another $PARSE using the same FAB
251 0249 2 ! on a null string to release these resources.
252 0250 2 !-
253 0251 2
254 0252 2 FAB[FAB$B_FNS] = 0;
255 0253 2 $PARSE(FAB=FAB);
256 0254 2 RETURN LIB$_INVFILSPE;
257 0255 2 END;
258 0256 2
259 0257 2 !+
260 0258 2 ! Perform a $SEARCH to get the file identification of the file.
261 0259 2 !-
262 0260 2
263 0261 2 STATUS_3 = $SEARCH(FAB=FAB);
264 0262 2 IF NOT .STATUS_3 THEN RETURN .STATUS_3;
265 0263 2
266 0264 2 !+
267 0265 2 ! Set up the FIB.
268 0266 2 !-
269 0267 2
270 0268 2 CH$FILL(0, FIB$C_LENGTH, FIB);
271 0269 2 FIB[FIB$ACCTL] = FIB$M_WRITE OR FIB$M_NOREAD OR FIB$M_NOWRITE;
272 0270 2 FIB[FIB$FID_NUM] = .NAM[NAM$FID_NUM];
273 0271 2 FIB[FIB$FID_SEQ] = .NAM[NAM$FID_SEQ];
274 0272 2 FIB[FIB$FID_RVN] = .NAM[NAM$FID_RVN];
275 0273 2
276 0274 2 !+
277 0275 2 ! Set up the FIB descriptor.
278 0276 2 !-
279 0277 2
280 0278 2 FIB_DESC[0] = FIB$C_LENGTH;
281 0279 2 FIB_DESC[1] = FIB;
282 0280 2
283 0281 2 !+
```



```

284 0282 2  | Set up the attribute list.
285 0283 2  | -
286 0284 2  |
287 0285 2  | ATR[0, ATR$W_TYPE] = ATR$C_UCHAR;      ! File characteristics
288 0286 2  | ATR[0, ATR$W_SIZE] = ATR$S_UCHAR;
289 0287 2  | ATR[0, ATR$L_ADDR] = UCHAR;
290 0288 2  | ATR[1, 0,0,32,0] = 0;                  ! End of list
291 0289 2  |
292 0290 2  | !+
293 0291 2  | | Assign a channel to the device.
294 0292 2  | | -
295 0293 2  | |
296 0294 2  | | TEMP_DESC[DSC$W_LENGTH] = .NAM[NAM$B_DEV];
297 0295 2  | | TEMP_DESC[DSC$A_POINTER] = .NAM[NAM$C_DEV];
298 0296 2  | | STATUS_4 = $ASSIGN(DEVNAM=TEMP_DESC, CHAN=CHANNEL);
299 0297 2  | | IF NOT .STATUS_4 THEN RETURN .STATUS_4;
300 0298 2  | |
301 0299 2  | | !+
302 0300 2  | | | Allocate an event flag.
303 0301 2  | | | -
304 0302 2  | | |
305 0303 2  | | | STATUS_5 = LIB$GET_EF(EFN);
306 0304 2  | | | IF NOT .STATUS_5
307 0305 2  | | | THEN
308 0306 2  | | |     BEGIN
309 0307 2  | | |     $DASSGN(CHAN=.CHANNEL);
310 0308 2  | | |     RETURN .STATUS_5;
311 0309 2  | | |     END;
312 0310 2  | | |
313 0311 2  | | | !+
314 0312 2  | | | | Beginning of block that is exited when processing is complete. FINAL_STATUS
315 0313 2  | | | | contains the status to be returned to caller.
316 0314 2  | | | | -
317 0315 2  | | | |
318 0316 2  | | | | PROCESS: BEGIN
319 0317 2  | | | |
320 0318 2  | | | | !+
321 0319 2  | | | | | Read at .ibytes of the file.
322 0320 2  | | | | | -
323 0321 2  | | | | |
324 0322 2  | | | | | FINAL STATUS = $QIOW(
325 0323 2  | | | | |     FUNC=IOS$ ACCESS,
326 0324 2  | | | | |     CHAN=.CHANNEL,
327 0325 2  | | | | |     EFN=.EFN,
328 0326 2  | | | | |     IOSB=IOSB,
329 0327 2  | | | | |     P1=FIB DESC,
330 0328 2  | | | | |     P5=ATR);
331 0329 2  | | | | | IF .FINAL STATUS THEN FINAL STATUS = .IOSB[0];
332 0330 2  | | | | | IF NOT .FINAL STATUS THEN LEAVE PROCESS;
333 0331 2  | | | | | FINAL_STATUS = SSS_WASCLR;
334 0332 2  | | | | |
335 0333 2  | | | | | !+
336 0334 2  | | | | | | If the file is currently locked, clear the lock bit.
337 0335 2  | | | | | | -
338 0336 2  | | | | | |
339 0337 2  | | | | | | IF TESTBITSC(UCHAR[FCH$V_LOCKED])
340 0338 2  | | | | | | THEN

```

P
P
P
P
P

MA
Sy
FM

PS
--
L

Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
17
Th
14
0

Ma
--
_S
O
Th
MA

```

341      0339 4      BEGIN
342      0340 4
343      0341 4      !+
344      0342 4      ! Rewrite the file characteristics to clear the lock.
345      0343 4      !-
346      0344 4
347      P 0345 4      FINAL_STATUS = $QIOW(
348      P 0346 4          FONC=IOS$ MODIFY,
349      P 0347 4          CHAN=.CHANNEL,
350      P 0348 4          EFN=.EFN,
351      P 0349 4          IOSB=IOSB,
352      P 0350 4          P1=FIB_DE$C,
353      0351 4          P5=ATR);
354      0352 4      IF .FINAL_STATUS THEN FINAL_STATUS = .IOSB[0];
355      0353 4      IF NOT .FINAL_STATUS THEN LEAVE PROCESS;
356      0354 4      FINAL_STATUS = $$$_WASSET;
357      0355 4      END;
358      0356 3
359      0357 3      !+
360      0358 3      ! End of block that is exited when processing is complete. FINAL_STATUS
361      0359 3      ! contains the status that is to be returned to caller.
362      0360 3      !-
363      0361 3
364      0362 2      END; ! of block PROCESS
365      0363 2
366      0364 2      !+
367      0365 2      ! Deassign the channel and deallocate the event flag.
368      0366 2      !-
369      0367 2
370      0368 2      STATUS_6 = $DASSGN(CHAN=.CHANNEL);
371      0369 2      STATUS_7 = LIB$FREE_EF(EFN);
372      0370 2      IF NOT .STATUS_7 THEN RETURN .STATUS_7;
373      0371 2      IF NOT .STATUS_6 THEN RETURN .STATUS_6;
374      0372 2
375      0373 2      !+
376      0374 2      ! Return the status.
377      0375 2      !-
378      0376 2
379      0377 2      RETURN .FINAL_STATUS;
380      0378 1      END;

```

! End of routine LIB\$UNLOCK_FILE

```

.TITLE LIB$UNLOCK_FILE LIB$UNLOCK_FILE - Clear deaccess
      s lock on file
.IDENT  \V04-000\
.EXTRN LIB$ANALYZE_SDESC R2
.EXTRN LIB$FREE_EF, LIB$GET_EF
.EXTRN LIB$INVARG, LIB$INVFILSPE
.EXTRN SY$$PARSE, SY$$SEARCH
.EXTRN SY$$ASSIGN, SY$$DASSGN
.EXTRN SY$$QIOW
.PSECT  _LIB$CODE, NOWRT, SHR, PIC, 2
.ENTRY  LIB$UNLOCK_FILE, Save R2,R3,R4,R5,R6,R7,R8,-; 0112
      R9,R10,R11

```

OFFC 00000

0050	8F	00	58	00000000G	00	9E	00002	MOVAB	SYSSQIOW, R8	0200		
			57	00000000G	00	9E	00009	MOVAB	SYSSDASSGN, R7	0207		
			56	00000000G	00	9E	00010	MOVAB	SYSSPARSE, R6			
			5E	FDDC	CE	9E	00017	MOVAB	-548(SP), SP			
					6C	95	0001C	TSTB	(AP)			
					4F	13	0001E	BEQL	1\$			
			6E		00	2C	00020	MOVCS	#0, (SP), #0, #80, \$RMS_PTR			
				B0	AD		00027					
			B0	AD	5003	8F	B0	00029	MOVW	#20483, \$RMS_PTR		
			C6	AD		02	90	0002F	MOVB	#2, \$RMS_PTR+22		
			CF	AD		02	90	00033	MOVB	#2, \$RMS_PTR+31		
0060	8F	00	D8	AD	FF50	CD	9E	00037	MOVAB	NAM, \$RMS_PTR+40	0210	
			6E			00	2C	0003D	MOVCS	#0, (SP), #C, #96, \$RMS_PTR		
				FF50	CD			00044				
				6002	8F	B0	00047	MOVW	#24578, \$RMS_PTR			
			FF5A	CD		01	8E	0004E	MNEGB	#1, \$RMS_PTR+10		
			FF5C	CD	74	AE	9E	00053	MOVAB	ESA_BUFFER, \$RMS_PTR+12		
			50		04	AC	D0	00059	MOVL	FILE_SPEC, R0	0221	
				00000000G	00	16	0005D	JSB	LIB\$ANALYZE_SDESC_R2			
			4A			50	E9	00063	BLBC	STATUS_1, 5\$	0222	
			000000FF	8F		51	D1	00066	CMPL	R1, #255	0223	
						08	1B	0006D	BLEQU	2\$		
			50	00000000G	8F	D0	0006F	1\$:	MOVL	#LIB\$_INVARG, R0		
						04	00076	RET				
			E4	AD		51	90	00077	2\$:	MOVB	R1, FAB+52	0224
			DC	AD		52	D0	0007B	MOVL	R2, FAB+44	0225	
					B0	AD	9F	0007F	PUSHAB	FAB	0232	
			66			01	FB	00082	CALLS	#1, SYSSPARSE		
			79			50	E9	00085	BLBC	STATUS_2, 6\$	0233	
			0201	8F	85	AD	B3	00088	BITW	NAM+52, #513	0241	
						05	12	0008E	BNEQ	3\$		
			11	F3	AD	04	E0	00090	BBS	#4, FAB+67, 4\$	0242	
					E4	AD	94	00095	3\$:	CLRB	FAB+52	0252
					B0	AD	9F	00098	PUSHAB	FAB	0253	
			66			01	FB	0009B	CALLS	#1, SYSSPARSE		
			50	00000000G	8F	D0	0009E	MOVL	#LIB\$_INVFILSPE, R0	0254		
						04	000A5	RET				
					B0	AD	9F	000A6	4\$:	PUSHAB	FAB	0261
			00000000G	00		01	FB	000A9	CALLS	#1, SYSSSEARCH		
			4E			50	E9	000B0	5\$:	BLBC	STATUS_3, 6\$	0262
0040	8F	00	6E			00	2C	000B3	MOVCS	#0, (SP), #0, #64, FIB	0268	
					28	AE		000BA				
			28	AE	0501	8F	3C	000BC	MOVZWL	#1281, FIB	0269	
			2C	AE	FF74	CD	D0	000C2	MOVL	NAM+36, FIB+4	0270	
			30	AE	FF78	CD	B0	000C8	MOVW	NAM+40, FIB+8	0272	
			20	AE	40	8F	9A	000CE	MOVZBL	#64, FIB_DESC	0278	
			24	AE	28	AE	9E	000D3	MOVAB	FIB, FIB_DESC+4	0279	
			10	AE	00030004	8F	D0	000D8	MOVL	#196612, ATR	0286	
			14	AE	68	AE	9E	000E0	MOVAB	UCHAR, ATR+4	0287	
					18	AE	D4	000E5	CLRL	ATR+8	0288	
			6C	AE	89	AD	9B	000E8	MOVZBW	NAM+57, TEMP_DESC	0294	
			70	AE	94	AD	D0	000ED	MOVL	NAM+68, TEMP_DESC+4	0295	
						7E	7C	000F2	CLRQ	-(SP)	0296	
					08	AE	9F	000F4	PUSHAB	CHANNEL		
					78	AE	9F	000F7	PUSHAB	TEMP_DESC		
			00000000G	00		04	FB	000FA	CALLS	#4, SYSSASSIGN		
				01		50	E8	00101	6\$:	BLBS	STATUS_4, 7\$	0297

		04	AE	04	00104	RET		
	00000000G		01	9F	00105	PUSHAB	EFN	0303
			50	FB	00108	CALLS	#1, LIB\$GET EF	
			52	DO	0010F	MOVL	R0, STATUS_5	
			52	E8	00112	BLBS	STATUS_5, 8\$	0304
			6E	3C	00115	MOVZWL	CHANNEL, -(SP)	0307
			01	FB	00118	CALLS	#1, SYSS\$DASSGN	
			7C	11	0011B	BRB	10\$	0308
			7E	D4	0011D	CLRL	-(SP)	0328
		14	AE	9F	0011F	PUSHAB	ATR	
			7E	7C	00122	CLRQ	-(SP)	
			7E	D4	00124	CLRL	-(SP)	
		34	AE	9F	00126	PUSHAB	FIB_DESC	
			7E	7C	00129	CLRQ	-(SP)	
		28	AE	9F	0012B	PUSHAB	IOSB	
			32	DD	0012E	PUSHL	#50	
		28	AE	3C	00130	MOVZWL	CHANNEL, -(SP)	
		30	AE	DD	00134	PUSHL	EFN	
			0C	FB	00137	CALLS	#12, SYSS\$QIOW	
			50	DO	0013A	MOVL	R0, FINAL_STATUS	
			52	E9	0013D	BLBC	FINAL_STATUS, 9\$	0329
		08	AE	3C	00140	MOVZWL	IOSB, FINAL_STATUS	
			52	E9	00144	BLBC	FINAL_STATUS, 9\$	0330
			01	DO	00147	MOVL	#1, FINAL_STATUS	0331
			06	E5	0014A	BBCC	#6, UCHAR, 9\$	0337
			7E	D4	0014F	CLRL	-(SP)	0351
		14	AE	9F	00151	PUSHAB	ATR	
			7E	7C	00154	CLRQ	-(SP)	
			7E	D4	00156	CLRL	-(SP)	
		34	AE	9F	00158	PUSHAB	FIB_DESC	
			7E	7C	0015B	CLRQ	-(SP)	
		28	AE	9F	0015D	PUSHAB	IOSB	
			36	DD	00160	PUSHL	#54	
		28	AE	3C	00162	MOVZWL	CHANNEL, -(SP)	
		30	AE	DD	00166	PUSHL	EFN	
			0C	FB	00169	CALLS	#12, SYSS\$QIOW	
			50	DO	0016C	MOVL	R0, FINAL_STATUS	
			52	E9	0016F	BLBC	FINAL_STATUS, 9\$	0352
		08	AE	3C	00172	MOVZWL	IOSB, FINAL_STATUS	
			52	E9	00176	BLBC	FINAL_STATUS, 9\$	0353
			09	DO	00179	MOVL	#9, FINAL_STATUS	0354
			6E	3C	0017C	MOVZWL	CHANNEL, -(SP)	0368
			01	FB	0017F	CALLS	#1, SYSS\$DASSGN	
			50	DO	00182	MOVL	R0, STATUS_6	
		04	AE	9F	00185	PUSHAB	EFN	0369
	00000000G		01	FB	00188	CALLS	#1, LIB\$FREE EF	
			50	E9	0018F	BLBC	STATUS_7, 11\$	0370
			53	E8	00192	BLBS	STATUS_6, 10\$	0371
			50	DO	00195	MOVL	STATUS_6, R0	
			04	00198	RET			
			52	DO	00199	MOVL	FINAL_STATUS, R0	0377
			04	0019C	RET			0378

; Routine Size: 413 bytes, Routine Base: _LIB\$CODE + 0000

: 382 0379 1 END
 : 383 0380 0 ELUDOM
 ! End of module LIB\$UNLOCK_FILE

PSECT SUMMARY

Name	Bytes	Attributes
_LIB\$CODE	413	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	91	0	1000	00:01.8

COMMAND QUALIFIERS

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

: Size: 413 code + 0 data bytes
 : Run Time: 00:12.1
 : Elapsed Time: 00:13.8
 : Lines/CPU Min: 1884
 : Lexemes/CPU-Min: 26434
 : Memory Used: 177 pages
 : Compilation Complete

