

1111111111	NNN	NNN	SSSSSSSSSSSS	TTTTTTTTTTTTTT	AAAAAAAAAA	LLL	
1111111111	NNN	NNN	SSSSSSSSSSSS	TTTTTTTTTTTTTT	AAAAAAAAAA	LLL	
1111111111	NNN	NNN	SSSSSSSSSSSS	TTTTTTTTTTTTTT	AAAAAAAAAA	LLL	
111	NNN	NNN	SSS	TTT	AAA	AAA	LLL
111	NNN	NNN	SSS	TTT	AAA	AAA	LLL
111	NNN	NNN	SSS	TTT	AAA	AAA	LLL
111	NNNNNN	NNN	SSS	TTT	AAA	AAA	LLL
111	NNNNNN	NNN	SSS	TTT	AAA	AAA	LLL
111	NNNNNN	NNN	SSS	TTT	AAA	AAA	LLL
111	NNN	NNN	NNN	SSSSSSSSS	AAA	AAA	LLL
111	NNN	NNN	NNN	SSSSSSSSS	AAA	AAA	LLL
111	NNN	NNN	NNN	SSSSSSSSS	AAA	AAA	LLL
111	NNN	NNNNNN	SSS	TTT	AAAAAAAAAAAA	LLL	
111	NNN	NNNNNN	SSS	TTT	AAAAAAAAAAAA	LLL	
111	NNN	NNNNNN	SSS	TTT	AAAAAAAAAAAA	LLL	
111	NNN	NNN	SSS	TTT	AAA	AAA	LLL
111	NNN	NNN	SSS	TTT	AAA	AAA	LLL
111	NNN	NNN	SSS	TTT	AAA	AAA	LLL
1111111111	NNN	NNN	SSSSSSSSSSSS	TTT	AAA	AAA	LLLLLLLLLLLLLLLL
1111111111	NNN	NNN	SSSSSSSSSSSS	TTT	AAA	AAA	LLLLLLLLLLLLLLLL
1111111111	NNN	NNN	SSSSSSSSSSSS	TTT	AAA	AAA	LLLLLLLLLLLLLLLL

```
IIIIII  NN    NN    SSSSSSSS  DDDDDDDD  EEEEEEEEE  LL    EEEEEEEEE  TTTTTTTTT  EEEEEEEEE
IIIIII  NN    NN    SSSSSSSS  DDDDDDDD  EEEEEEEEE  LL    EEEEEEEEE  TTTTTTTTT  EEEEEEEEE
  II    NN    NN    SS        DD        EE        LL    EE        TT        EE
  II    NN    NN    SS        DD        EE        LL    EE        TT        EE
  II    NNNN  NN    SS        DD        EE        LL    EE        TT        EE
  II    NNNN  NN    SS        DD        EE        LL    EE        TT        EE
  II    NN    NN    SSSSSS    DD        EE        LL    EEEEEEE  TT        EEEEEEE
  II    NN    NN    SSSSSS    DD        EE        LL    EEEEEEE  TT        EEEEEEE
  II    NN    NN    SS        DD        EE        LL    EE        TT        EE
  II    NN    NNNN  SS        DD        EE        LL    EE        TT        EE
  II    NN    NNNN  SS        DD        EE        LL    EE        TT        EE
  II    NN    NN    SS        DD        EE        LL    EE        TT        EE
  II    NN    NN    SSSSSS    DD        EEEEEEEEE  LL    EEEEEEEEE  TT        EEEEEEEEE
IIIIII  NN    NN    SSSSSSSS  DDDDDDDD  EEEEEEEEE  LLLLLLLLLL  EEEEEEEEE  TT        EEEEEEEEE
IIIIII  NN    NN    SSSSSSSS  DDDDDDDD  EEEEEEEEE  LLLLLLLLLL  EEEEEEEEE  TT        EEEEEEEEE
```

```
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 MODULE INSDELETE ( ! DELETE a KFE entry
2 0002 0 IDENT = 'V04-000';
3 0003 0 ADDRESSING_MODE(EXTERNAL = GENERAL)
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 FACILITY: Install
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module executes the REPLACE and DELETE options on INSTALL
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1
40 0040 1 VAX/VMS operating system.
41 0041 1
42 0042 1 AUTHOR: Bob Grosso, April 1983
43 0043 1
44 0044 1 Modified by:
45 0045 1
46 0046 1 V03-012 MSH0061 Michael S. Harvey 5-Jul-1984
47 0047 1 Add EXEONLY support.
48 0048 1
49 0049 1 V03-011 MSH0057 Michael S. Harvey 26-Jun-1984
50 0050 1 Propagate WRITEABLE attribute across REPLACE command.
51 0051 1
52 0052 1 V03-010 MSH0050 Michael S. Harvey 18-May-1984
53 0053 1 Convert match control to MATEQU before deleting global
54 0054 1 sections. This ensures that DELETE/REMOVE doesn't
55 0055 1 delete newly created global sections of the same name
56 0056 1 but with different idents. In other words, we only
57 0057 1 want to delete global sections that are likely to have
```

```

58      0058      1      |
59      0059      1      |
60      0060      1      |
61      0061      1      |
62      0062      1      |
63      0063      1      |
64      0064      1      |
65      0065      1      |
66      0066      1      |
67      0067      1      |
68      0068      1      |
69      0069      1      |
70      0070      1      |
71      0071      1      |
72      0072      1      |
73      0073      1      |
74      0074      1      |
75      0075      1      |
76      0076      1      |
77      0077      1      |
78      0078      1      |
79      0079      1      |
80      0080      1      |
81      0081      1      |
82      0082      1      |
83      0083      1      |
84      0084      1      |
85      0085      1      |
86      0086      1      |
87      0087      1      |
88      0088      1      |
89      0089      1      |
90      0090      1      |
91      0091      1      |
92      0092      1      |
93      0093      1      |
94      0094      1      |
95      0095      1      |
96      0096      1      |
97      0097      1      |
98      0098      1      |
99      0099      1      |
100     0100      1      | Include files
101     0101      1      |
102     0102      1      |
103     0103      1      | LIBRARY 'SYSS$LIBRARY:LIB';
104     0104      1      |
105     0105      1      | REQUIRE 'SRC$:INSPREFIX.REQ';
106     0247      1      | REQUIRE 'LIB$:INSDEF.R32';

      been created at the time the about-to-be-deleted image
      was created, not any global sections of the same name
      that have been created since that time.

V03-009 MSH0027      Michael S. Harvey      6-Apr-1984
      Ensure that files get closed during the PURGE operation.

V03-008 MSH0022      Michael S. Harvey      19-Mar-1984
      Convert and compress directory brackets correctly.

V03-007 MSH0017      Michael S. Harvey      7-Mar-1984
      Convert FIND_KFE into a global routine for use during
      known file creation.

V03-006 MSH0003      Michael S. Harvey      27-Jan-1984
      Prevent crash from executing system service at elevated
      IPL by resetting IPL on wierd error paths, such as is
      done on success paths.

V03-005 RPG0005      Bob Grosso      08-Sep-1983
      Raise IPL to protect memory deallocation, so a process
      won't result in pool loss.
      Do not decrement REFCNT in WCB when deleting entry.

V03-004 RPG0004      Bob Grosso      16-Aug-1983
      Fix /REPLACE so it retains attributes.
      Activate REMOVE command.

V03-003 RPG0003      Bob Grosso      July 25, 1983
      Decrement entry count upon deletion.
      Clean up PURGE.
      Clean up error reporting.

V03-002 RPG0002      Bob Grosso      July 19, 1983
      Do not decrement REFCNT in WCB when deleting entry.

V03-001 RPG0001      Bob Grosso      July 7, 1983
      Take out exclusive lock when purging

```


Declarations

```
108 0306 1 %SBTTL 'Declarations';
109 0307 1
110 0308 1 LINKAGE
111 0309 1     CALL_S_4 = CALL (STANDARD, REGISTER=4),           ! for calling routines to execute in kernel mode,
112 0310 1                                           ! the change mode dispatcher passes PCB address in R
113 0311 1
114 0312 1     JSB_0 = JSB (REGISTER = 0),                       ! Deallocate paged pool
115 0313 1
116 0314 1     JSB_0_1_2 = JSB (REGISTER = 0; REGISTER = 1, REGISTER = 2) ! Verify I/O channel
117 0315 1         : NOPRESERVE (3)
118 0316 1         : NOTUSED (4,5,6,7,8,9,10,11),
119 0317 1
120 0318 1     JSB_G1_G2_3 = JSB (REGISTER = 3)                   ! Allocate memory in P1 space
121 0319 1         : GLOBAL (length = 1, entry_block = 2);
122 0320 1
123 0321 1 !
124 0322 1 ! Table of contents
125 0323 1 !
126 0324 1
127 0325 1 EXTERNAL
128 0326 1     ctl$gq_allocreg,           ! Memory allocation listhead
129 0327 1     ctl$gl_knownfil,         ! Process known file listhead queues
130 0328 1     EXE$GL_KNOWN_FILES : REF BBLOCK, ! Pointer to knownfil list queues
131 0329 1     SGN$GB_KFHSH$IZ;          ! Number of buckets in KF hash table
132 0330 1
133 0331 1 EXTERNAL ROUTINE
134 0332 1     IOC$VERIFYCHAN : JSB_0_1_2,
135 0333 1     EXE$DEAPAGED : JSB_0,
136 0334 1     SYSS$ASSIGN,
137 0335 1     SYSS$DASSGN;
138 0336 1
139 0337 1
140 0338 1 EXTERNAL LITERAL
141 0339 1     INSS$_EMPTYLIST,           ! The Known File List is empty
142 0340 1     INSS$_INTRNLERR,         ! no known file entry found
143 0341 1     INSS$_NOKFEFND,          ! no known file entry found during REMOVE
144 0342 1     INSS$_NOLIST,            ! There is no Known File list
145 0343 1     INSS$_REMOVED;          ! Dangling entry was removed
146 0344 1
147 0345 1 EXTERNAL
148 0346 1     INSS$GL_CTLMSK : BLOCK [1], ! Control flags
149 0347 1     INSS$GL_REPLACE_MSK : BLOCK [1], ! Control flags for REPLACE
150 0348 1     INSS$L_INTRNLERR,         ! Pass back internal error descriptor
151 0349 1     INSS$GL_KFECHAN,          ! Channel known image file is open on.
152 0350 1     INSS$G_RFENAM : $NAM_DECL, ! NAM block for the filename of the known image
153 0351 1     INSS$G_KFEPRIVS,          ! Privilege mask, save it if REPLACE
154 0352 1     INSS$GL_KFEADR;          ! Return the KFE address when it has been replaced
155 0353 1
156 0354 1 EXTERNAL ROUTINE
157 0355 1     INSS$EXECUTE IN KRNL_WITH_W_LOCK,
158 0356 1     INSS$CNVRT_KF_LOCK,
159 0357 1     INSS$BLD_GBLSECNAM,
160 0358 1     INSS$CVT_DIR,
161 0359 1     INSS$HASH;
162 0360 1
163 0361 1 FORWARD ROUTINE
164 0362 1     INSS$FIND_KFE,
```


INSDELETE
V04-000

Declarations

I 1
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 4
(2)

```
: 165      0363 1      INS_DELETE,  
: 166      0364 1      INS_PURGE,  
: 167      0365 1      PURGE,  
: 168      0366 1      DETACH_KFE,  
: 169      0367 1      RETURN_KFE,  
: 170      0368 1      DEALLOC_PAGED;  
: 171      0369 1  
: 172      0370 1 BIND  
: 173      0371 1      SGN_B_KFHSHSIZ = SGN$GB_KFHSHSIZ : BYTE;  
: 174      0372 1  
: 175      0373 1 BIND  
: 176      0374 1      HSHKFE_ERR_DSC = $DESCRIPTOR (' KFE for deletion not found in hash'),  
: 177      0375 1      IMGHDR_ERR_DSC = $DESCRIPTOR (' No image header found for delete'),  
: 178      0376 1      KFD_ERR_DSC = $DESCRIPTOR (' KFD for deletion not found'),  
: 179      0377 1      KFDKFE_ERR_DSC = $DESCRIPTOR (' KFE for deletion not found in KFD list'),  
: 180      0378 1      EMPTYKFPB_ERR_DSC = $DESCRIPTOR (' Deletion from empty pointer block');
```

```
! Remove entry from list  
! Deallocate entry  
! Deallocate paged pool
```



```
182 0379 1 %SBTTL 'INS$DELETE';
183 0380 1
184 0381 1 GLOBAL ROUTINE INS$DELETE =
185 0382 2 BEGIN
186 0383 2 +++
187 0384 2
188 0385 2 FUNCTIONAL DESCRIPTION:
189 0386 2
190 0387 2 Delete a Known File entry.
191 0388 2 If this was the last entry for the listhead, delete it
192 0389 2
193 0390 2 EXPLICIT INPUT:
194 0391 2
195 0392 2 none
196 0393 2
197 0394 2 ROUTINE VALUE:
198 0395 2
199 0396 2 R0 = return status, low bit set for success, else error status
200 0397 2
201 0398 2 ---
202 0399 2
203 0400 2 LOCAL
204 0401 2 STATUS;
205 0402 2
206 0403 2 IF .INS$GL_CTLMSK [INS$V_PROCESS] THEN RETURN TRUE;
207 0404 2
208 0405 2 STATUS = INS$EXECUTE_IN_KRNL_WITH_W_LOCK (INS_DELETE, 0);
209 0406 2 RETURN .STATUS;
210 0407 1 END; ! Global routine INS$DELETE
```

```
.TITLE INSDELETE
.IDENT \V04-000\

.PSECT $PLIT$,NOWRT,NOEXE,2

69 74 65 6C 65 64 20 72 6F 66 20 45 46 4B 20 00000 P.AAB: .ASCII \ KFE for deletion not found in hash\
6E 69 20 64 6E 75 6F 66 20 74 6F 6E 20 6E 6F 0000F
68 73 61 68 20 0001E
00023
00000023 00024 P.AAA: .BLKB 1
00000000 00028 .LONG 35
0002C P.AAD: .ADDRESS P.AAB
65 64 61 65 68 20 65 67 61 6D 69 20 6F 4E 20 0002C P.AAD: .ASCII \ No image header found for delete\
6C 65 64 20 72 6F 66 20 64 6E 75 6F 66 20 72 0003B
65 74 65 0004A
0004D
00000021 00050 P.AAC: .BLKB 3
00000000 00054 .LONG 33
00058 P.AAF: .ADDRESS P.AAD
69 74 65 6C 65 64 20 72 6F 66 20 44 46 4B 20 00058 P.AAF: .ASCII \ KFD for deletion not found\
64 6E 75 6F 66 20 74 6F 6E 20 6E 6F 00067
00073
0000001B 00074 P.AAE: .BLKB 1
00000000 00078 .LONG 27
69 74 65 6C 65 64 20 72 6F 66 20 45 46 4B 20 0007C P.AAH: .ADDRESS P.AAF
6E 69 20 64 6E 75 6F 66 20 74 6F 6E 20 6E 6F 0007C P.AAH: .ASCII \ KFE for deletion not found in KFD list\
74 73 69 6C 20 44 46 4B 20 0008B
0009A
000A3 .BLKB 1
```


INSDELETE
V04-000

INSSDELETE

K 1
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 6
(3)

20 6D 6F 72 66 20 6E 6F 69 74 65 6C 65 44 20
62 20 72 65 74 6E 69 6F 70 20 79 74 70 6D 65
6B 63 6F 6C
00000027, 000A4 P.AAG: .LONG 39
00000000, 000A8 .ADDRESS P.AAH
000AC P.AAJ: .ASCII \ Deletion from empty pointer block\
000BB
000CA
000CE
00000022, 000D0 P.AAI: .BLKB 2
00000000, 000D4 .LONG 34
000D4 .ADDRESS P.AAJ

MSHKFE_ERR_DSC= P.AAA
IMGHDR_ERR_DSC= P.AAC
KFD_ERR_DSC= P.AAE
KFDKFE_ERR_DSC= P.AAG
EMPTYKFPB_ERR_DSC= P.AAI
.EXTRN CTL\$GQ_ALLOCREG
.EXTRN CTL\$GL_KNOWNFIL
.EXTRN EXE\$GL_KNOWN_FILES
.EXTRN SGN\$GB_KFHSHSIZ
.EXTRN LOC\$VERIFYCHAN, EXE\$DEAPAGED
.EXTRN SYSS\$ASSIGN, SYSS\$DASSGN
.EXTRN INSS_EMPTYLIST, INSS_INTRNLERR
.EXTRN INSS_NOKFEFND, INSS_NOLIST
.EXTRN INSS_REMOVED, INSSGL_CTLMSK
.EXTRN INSSGL_REPLACE_MSK
.EXTRN INSSL_INTRNLERR
.EXTRN INSSGL_KFECHAN, INSSG_KFENAM
.EXTRN INSSGQ_KFEPRIVS
.EXTRN INSSGL_KFEADR, INS\$EXECUTE_IN_KRNL_WITH_W_LOCK
.EXTRN INSSCNVRT_KF_LOCK
.EXTRN INSSBLD_GBLSECNAM
.EXTRN INSSCVT_DIR, INSSHASH

.PSECT \$CODE\$,NOWRT,2

04 00000000G 00 0000 00000
50 01 E1 00002
01 D0 0000A
04 0000D
7E D4 0000E 1\$:
0000V CF 9F 00010
00000000G 00 02 FB 00014
04 0001B

.ENTRY INSSDELETE, Save nothing
BBC #1, INSSGL_CTLMSK, 1\$
MOVL #1, R0
RET
CLRL -(SP)
PUSHAB INS_DELETE
CALLS #2, INS\$EXECUTE_IN_KRNL_WITH_W_LOCK
RET

: 0381
: 0403
: 0405
: 0407

; Routine Size: 28 bytes, Routine Base: \$CODE\$ + 0000

; 211 0408 1


```
213 0409 1 %SBTTL 'INS_DELETE';
214 0410 1
215 0411 1 GLOBAL ROUTINE INS_DELETE =
216 0412 2 BEGIN
217 0413 2 +++
218 0414 2
219 0415 2 FUNCTIONAL DESCRIPTION:
220 0416 2
221 0417 2 Delete a Known File entry.
222 0418 2 If this was the last entry for the listhead, delete it
223 0419 2
224 0420 2 EXPLICIT INPUT:
225 0421 2
226 0422 2 none
227 0423 2
228 0424 2 IMPLICIT INPUT:
229 0425 2
230 0426 2 INSSGL_KFEADR = Channel on which the known file image is open
231 0427 2
232 0428 2 IMPLICIT OUTPUT:
233 0429 2
234 0430 2 INSSGL_KFEADR = Address of KFE, will be set to zero
235 0431 2
236 0432 2 ROUTINE VALUE:
237 0433 2
238 0434 2 R0 = return status, low bit set for success, else error status
239 0435 2
240 0436 2 ---
241 0437 2
242 0438 2 LOCAL
243 0439 2 HASH_INDEX,
244 0440 2 KFD : REF BBLOCK,
245 0441 2 KFE,
246 0442 2 STATUS;
247 0443 2
248 0444 2
249 0445 2 If this is a REMOVE, i.e. We suspect there is a dangling entry,
250 0446 2 then compute which hash table bucket Known File Entry should be in.
251 0447 2 If we find the entry, remove it, else return normally.
252 0448 2
253 0449 2 IF .INSSGL_KFEADR EQL 0
254 0450 2 THEN
255 0451 2 BEGIN
256 0452 2 HASH_INDEX = INSSHASH (.INSSG KFENAM [NAM$B NAME],
257 0453 2 .INSSG_KFENAM [NAM$L_NAME], .SGN_B_KFHSHSIZ );
258 0454 2
259 0455 2 KFE = INSFIND_KFE (.HASH_INDEX, INSSG_KFENAM);
260 0456 2
261 0457 2 IF .KFE EQL 0 THEN RETURN INSS_NOKFEFND;
262 0458 2
263 0459 2 INSSCNVRT_KF_LOCK (LCK$K_EXMODE); ! Convert protected read to exclusive
264 0460 2 ! to lock out any image activations
265 0461 2 SET IPL (IPL$ASTDEL); ! Block deletion of process so pool is not lost
266 0462 2 STATUS = DETACH_KFE (.KFE, KFD);
267 0463 2 IF .KFD NEQ 0
268 0464 2 THEN
269 0465 2 BEGIN
```

```

: 270      0466      4      STATUS = DEALLOC_PAGED (.KFD);
: 271      0467      4      END;
: 272      0468      4
: 273      0469      4      STATUS = RETURN_KFE (.KFE,0);
: 274      0470      4
: 275      0471      4      SET IPL (0);
: 276      0472      4      INSSCNVRT_KF_LOCK (LCK$K_PMODE);
: 277      0473      4      RETURN INSS_REMOVED;
: 278      0474      4      END;
: 279      0475      4
: 280      0476      4
: 281      0477      4
: 282      0478      4      !
: 283      0479      4      ! Take out write lock and remove entry from list
: 284      0480      4
: 285      0481      4      INSSCNVRT_KF_LOCK (LCK$K_EXMODE);
: 286      0482      4      ! Convert protected read to exclusive
: 287      0483      4      SET_IPL (IPL$ASTDEL);
: 288      0484      4      ! Block deletion of process so pool is not lost
: 289      0485      4      STATUS = DETACH_KFE (.INSSGL_KFEADR, KFD);
: 290      0486      4      IF .STATUS
: 291      0487      4      THEN
: 292      0488      4      BEGIN
: 293      0489      4      IF .KFD NEQ 0
: 294      0490      4      THEN
: 295      0491      4      STATUS = DEALLOC_PAGED (.KFD);
: 296      0492      4
: 297      0493      4      !
: 298      0494      4      ! Clean up entry and deallocate it
: 299      0495      4      !
: 300      0496      4      STATUS = RETURN_KFE (.INSSGL_KFEADR,0);
: 301      0497      4      INSSGL_KFEADR = 0;
: 302      0498      4      END;
: 303      0499      4
: 304      0500      4      SET IPL (0);
: 305      0501      4      INSSCNVRT_KF_LOCK (LCK$K_PMODE);
: 306      0502      4      ! Convert exclusive to protected read
: 307      0503      4      ! to allow image activations
: 308      0504      4      RETURN .STATUS;
: 309      0505      1      END;
:                               ! Global routine INS_DELETE
```

```

                                007C 00000
56 00000000G 00 9E 00002
55 00000000G 00 9E 00009
54 00000000G 00 9E 00010
5E          04 C2 00017
          65 D5 0001A
          69 12 0001C
7E 00000000G 00 9A 0001E
          66 DD 00025
7E          EF A6 9A 00027
00000000G 00 03 FB 0002B
          B4 A6 9F 00032
```

```

.ENTRY  INS_DELETE, Save R2,R3,R4,R5,R6
MOVAB   INSSG_KFENAM+76, R6
MOVAB   INSSGL_KFEADR, R5
MOVAB   INSSCNVRT_KF_LOCK, R4
SUBL2   #4, SP
TSTL    INSSGL_KFEADR
BNEQ    3$
MOVZBL  SGN_B_KFHSHSIZ, -(SP)
PUSHL   INSSG_KFENAM+76
MOVZBL  INSSG_KFENAM+59, -(SP)
CALLS   #3, INSSHASH
PUSHAB  INSSG_KFENAM
```

```

: 0411
:
:
:
: 0449
:
: 0453
:
: 0452
:
: 0455
```


INSDELETE
V04-000

INS_DELETE

N 1
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 9
(4)

0000V	CF	50	DD	00035	PUSHL	HASH_INDEX	
	52	02	FB	00037	CALLS	#2, INSSFIND_KFE	
		50	D0	0003C	MOVL	R0, KFE	
		08	12	0003F	BNEQ	1\$	0457
	50	8F	D0	00041	MOVL	#INS\$_NOKFEFND, R0	
			04	00048	RET		
		05	DD	00049	PUSHL	#5	0459
	64	01	FB	0004B	CALLS	#1, INSSCNVRT_KF_LOCK	
	12	02	DA	0004E	MTPR	#2, #18	0461
		8F	BB	00051	PUSHR	#^M<R2, SP>	0462
0000V	CF	02	FB	00055	CALLS	#2, DETACH_KFE	
	53	50	D0	0005A	MOVL	R0, STATUS	
		6E	D5	0005D	TSTL	KFD	0463
		0A	13	0005F	BEQL	2\$	
		6E	DD	00061	PUSHL	KFD	0466
0000V	CF	01	FB	00063	CALLS	#1, DEALLOC_PAGED	
	53	50	D0	00068	MOVL	R0, STATUS	
		7E	D4	0006B	CLRL	-(SP)	0469
		52	DD	0006D	PUSHL	KFE	
0000V	CF	02	FB	0006F	CALLS	#2, RETURN_KFE	
	53	50	D0	00074	MOVL	R0, STATUS	
	12	00	DA	00077	MTPR	#0, #18	0471
		03	DD	0007A	PUSHL	#3	0472
	64	01	FB	0007C	CALLS	#1, INSSCNVRT_KF_LOCK	
	50	8F	D0	0007F	MOVL	#INS\$_REMOVED, R0	0474
			04	00086	RET		
		05	DD	00087	PUSHL	#5	0481
	64	01	FB	00089	CALLS	#1, INSSCNVRT_KF_LOCK	
	12	02	DA	0008C	MTPR	#2, #18	0483
		5E	DD	0008F	PUSHL	SP	0485
		65	DD	00091	PUSHL	INSSGL_KFEADR	
0000V	CF	02	FB	00093	CALLS	#2, DETACH_KFE	
	53	50	D0	00098	MOVL	R0, STATUS	
	1C	53	E9	0009B	BLBC	STATUS, 5\$	0486
		6E	D5	0009E	TSTL	KFD	0489
		0A	13	000A0	BEQL	4\$	
		6E	DD	000A2	PUSHL	KFD	0491
0000V	CF	01	FB	000A4	CALLS	#1, DEALLOC_PAGED	
	53	50	D0	000A9	MOVL	R0, STATUS	
		7E	D4	000AC	CLRL	-(SP)	0496
		65	DD	000AE	PUSHL	INSSGL_KFEADR	
0000V	CF	02	FB	000B0	CALLS	#2, RETURN_KFE	
	53	50	D0	000B5	MOVL	R0, STATUS	
		65	D4	000B8	CLRL	INSSGL_KFEADR	0497
	12	00	DA	000BA	MTPR	#0, #18	0500
		03	DD	000BD	PUSHL	#3	0501
	64	01	FB	000BF	CALLS	#1, INSSCNVRT_KF_LOCK	
	50	53	D0	000C2	MOVL	STATUS, R0	0504
			04	000C5	RET		0505

; Routine Size: 198 bytes, Routine Base: \$CODE\$ + 001C

; 310 0506 1


```

: 312 0507 1 %SBTTL 'INSSPURGE';
: 313 0508 1
: 314 0509 1 GLOBAL ROUTINE INSSPURGE =
: 315 0510 2 BEGIN
: 316 0511 2 +++
: 317 0512 2
: 318 0513 2 FUNCTIONAL DESCRIPTION:
: 319 0514 2
: 320 0515 2 Purge all known file entries that are purgeable.
: 321 0516 2 For each that is the last entry for the listhead, delete it
: 322 0517 2
: 323 0518 2 EXPLICIT INPUT:
: 324 0519 2
: 325 0520 2 none
: 326 0521 2
: 327 0522 2 ROUTINE VALUE:
: 328 0523 2
: 329 0524 2 R0 = return status, low bit set for success, else error status
: 330 0525 2
: 331 0526 2 ---
: 332 0527 2 LOCAL
: 333 0528 2 STATUS;
: 334 0529 2
: 335 0530 2 IF .INSSGL_CTLMSK [INSSV_PROCESS] THEN RETURN TRUE;
: 336 0531 2
: 337 0532 2 STATUS = INSS$EXECUTE_IN_KRNL_WITH_W_LOCK (INS_PURGE, 0);
: 338 0533 2 RETURN .STATUS;
: 339 0534 1 END; ! Global routine INSSPURGE

```

04 00000000G 00	01 E1 00002	.ENTRY INSSPURGE, Save nothing	: 0509
50	01 D0 0000A	BBC #1, INSSGL_CTLMSK, 1\$	: 0530
	04 0000D	MOVL #1, R0	:
	7E D4 0000E 1\$:	RET	:
	CF 9F 00010	CLRL -(SP)	: 0532
00000000G 00	02 FB 00014	PUSHAB INS_PURGE	:
0000V	04 0001B	CALLS #2, INSS\$EXECUTE_IN_KRNL_WITH_W_LOCK	: 0534
		RET	:

; Routine Size: 28 bytes, Routine Base: \$CODE\$ + 00E2

; 340 0535 1


```
INS_PURGE
0536 1 %SBTTL 'INS_PURGE';
0537 1
0538 1 ROUTINE INS_PURGE =
0539 2 BEGIN
0540 2 +++
0541 2
0542 2 FUNCTIONAL DESCRIPTION:
0543 2
0544 2 Purge all Known File entries that are purgeable.
0545 2 For each that is the last entry for the listhead, delete it
0546 2
0547 2 EXPLICIT INPUT:
0548 2
0549 2 none
0550 2
0551 2 IMPLICIT INPUT:
0552 2
0553 2 INSSGL_KFEADR = Channel on which the known file image is open
0554 2
0555 2 IMPLICIT OUTPUT:
0556 2
0557 2 INSSGL_KFEADR = Address of KFE, will be set to zero
0558 2
0559 2 ROUTINE VALUE:
0560 2
0561 2 R0 = return status, low bit set for success, else error status
0562 2
0563 2 ---
0564 2 LOCAL
0565 2 RET_STATUS,
0566 2 STATUS;
0567 2
0568 2
0569 2 BIND
0570 2 KFPB = EXESGL_KNOWN_FILES : REF BBLOCK;
0571 2
0572 2 IF .KFPB EQL 0
0573 2 THEN
0574 2 RETURN INSS_NOLIST;
0575 2
0576 2 IF .KFPB [KFPB$L_KFDLST] EQL 0
0577 2 THEN
0578 2 RETURN INSS_EMPTYLST;
0579 2
0580 2
0581 2 STATUS = INSSCNVRT_KF_LOCK (LCK$K_EXMODE); ! Convert protected read to exclusive
0582 2 ! to lock out any image activations
0583 2 IF NOT .STATUS THEN RETURN .STATUS;
0584 2
0585 2 RET_STATUS = PURGE ();
0586 2
0587 2 INSSCNVRT_KF_LOCK (LCK$K_PMODE); ! Convert exclusive to protected read
0588 2 ! to allow image activations
0589 2 IF NOT .STATUS THEN RETURN .STATUS;
0590 2
0591 2 INSSGL_KFEADR = 0;
0592 2
0593 2
0594 2
0595 2
0596 2
0597 2
0598 2
```

INSDELETE
V04-000

INS_PURGE

D 2
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 12
(6)

; 399
; 400

0593 2 RETURN .RET_STATUS;
0594 1 END; ! Global routine INS_PURGE

```
001C 00000 INS_PURGE:
54 00000000G 00 9E 00002 .WORD Save R2,R3,R4 ; 0538
50 00000000G 00 D0 00009 MOVAB INSSCNVRT_KF_LOCK, R4 ; 0572
08 12 00010 MOVL KFPB, R0 ; 0574
50 00000000G 8F D0 00012 MOVL #INSS_NOLIST, R0 ; 0576
04 00019 RET (R0) ; 0578
60 D5 0001A 1$: BNEQ 2$ ; 0581
08 12 0001C MOVL #INSS_EMPTYLIST, R0 ; 0583
50 00000000G 8F D0 0001E RET ; 0585
04 00025 PUSHL #5 ; 0587
64 01 FB 00028 CALLS #1, INSSCNVRT_KF_LOCK ; 0589
52 50 D0 0002B MOVL R0, STATUS ; 0591
10 52 E9 0002E BLBC STATUS, 3$ ; 0593
CF 00 FB 00031 CALLS #0, PURGE ; 0594
53 50 D0 00036 MOVL R0, RET_STATUS
03 DD 00039 PUSHL #3
64 01 FB 0003B CALLS #1, INSSCNVRT_KF_LOCK
04 52 E8 0003E BLBS STATUS, 4$
50 52 D0 00041 3$: MOVL STATUS, R0
04 00044 RET
0000V 00000000G 00 D4 00045 4$: CLRL INSSGL_KFEADR
50 53 D0 0004B MOVL RET_STATUS, R0
04 0004E RET
```

; Routine Size: 79 bytes, Routine Base: \$CODE\$ + 00FE

; 401 0595 1


```

: 403      0596 1 %SBTTL 'PURGE';
: 404      0597 1
: 405      0598 1 ROUTINE PURGE =
: 406      0599 2 BEGIN
: 407      0600 2 !+++
: 408      0601 2
: 409      0602 2 FUNCTIONAL DESCRIPTION:
: 410      0603 2
: 411      0604 2 Purge all Known File entries that are purgeable.
: 412      0605 2 For each that is the last entry for the listhead, delete it
: 413      0606 2
: 414      0607 2 EXPLICIT INPUT:
: 415      0608 2
: 416      0609 2
: 417      0610 2 ---
: 418      0611 2 LOCAL
: 419      0612 2 DEV_DSC : $BBLOCK [DSC$C_S_BLN],
: 420      0613 2 CHAN,
: 421      0614 2 CCB : REF $BBLOCK,
: 422      0615 2 KFD : REF BBLOCK,
: 423      0616 2 KFE : REF BBLOCK,
: 424      0617 2 RET_KFE : REF BBLOCK,
: 425      0618 2 RET_KFD,
: 426      0619 2 STATUS;
: 427      0620 2
: 428      0621 2
: 429      0622 2 BIND
: 430      0623 2 KFPB = EXE$GL_KNOWN_FILES : REF BBLOCK;
: 431      0624 2
: 432      0625 2 KFD = .KFPB [KFPB$L_KFDLST];
: 433      0626 2
: 434      0627 2 !
: 435      0628 2 Traverse the list of KFDs and access all its KFEs.
: 436      0629 2 The KFD is the header block which contains the Device, directory and
: 437      0630 2 file type which several Known File Entries (KFE) share in common.
: 438      0631 2
: 439      0632 2 WHILE .KFD NEQ 0 DO
: 440      0633 2 BEGIN
: 441      0634 2 KFE = .KFD [KFD$L_KFELIST];
: 442      0635 2
: 443      0636 2 Set up device string descriptor for assignment.
: 444      0637 2
: 445      0638 2 DEV_DSC [DSC$W_LENGTH] = .KFD [KFD$B_DEVLEN];
: 446      0639 2 DEV_DSC [DSC$A_POINTER] = KFD [KFD$T-DDTSTR];
: 447      0640 2 KFD = .KFD [KFD$L_LINK]; ! Follow link now to next KFD
: 448      0641 2
: 449      0642 2 DELETE each KFE in the KFD's ordered list of KFEs
: 450      0643 2
: 451      0644 2 WHILE .KFE NEQ 0 DO
: 452      0645 2 BEGIN
: 453      0646 2 RET_KFE = .KFE; ! KFE to be deleted
: 454      0647 2 KFE = .KFE [KFE$L_KFELINK]; ! Follow link now for next one
: 455      0648 2
: 456      0649 2 IF NOT .RET_KFE [KFE$V_NOPURGE]
: 457      0650 2 THEN
: 458      0651 2 BEGIN
: 459      0652 2 !
```

```

: 460      0653      5      | Get ready to close the file. Do this now before the data
: 461      0654      5      | structures get cleaned up at elevated IPL.
: 462      0655      5
: 463      0656      5      SYSS$ASSIGN (DEV DSC,CHAN,0,0);
: 464      0657      5      IOC$VERIFYCHAN T.CHAN; CCB);
: 465      0658      5
: 466      0659      5      | Take out write lock and remove entry from list
: 467      0660      5
: 468      0661      5      SET_IPL (IPL$_ASTDEL);      ! Block deletion of process so pool is not lost
: 469      0662      5
: 470      0663      5      STATUS = DETACH_KFE (.RET_KFE, RET_KFD);
: 471      0664      5
: 472      0665      5      IF NOT .STATUS
: 473      0666      5      THEN
: 474      0667      6      BEGIN
: 475      0668      6      SET_IPL (0);
: 476      0669      6      RETURN .STATUS;
: 477      0670      5      END;
: 478      0671      5
: 479      0672      5      IF .RET_KFD NEQ 0
: 480      0673      5      THEN
: 481      0674      6      BEGIN
: 482      0675      6      STATUS = DEALLOC_PAGED (.RET_KFD);
: 483      0676      6
: 484      0677      6      IF NOT .STATUS
: 485      0678      6      THEN
: 486      0679      7      BEGIN
: 487      0680      7      SET_IPL (0);
: 488      0681      7      RETURN .STATUS;
: 489      0682      6      END;
: 490      0683      5      END;
: 491      0684      5
: 492      0685      5      | Clean up entry and deallocate it
: 493      0686      5
: 494      0687      5      STATUS = RETURN_KFE (.RET_KFE, .CCB);
: 495      0688      5      SET_IPL (0);
: 496      0689      5
: 497      0690      5      | Now, prepare for final file closing
: 498      0691      5
: 499      0692      5      SYSS$DASSGN (.CHAN);
: 500      0693      5
: 501      0694      5      IF NOT .STATUS THEN RETURN .STATUS;
: 502      0695      5      END;
: 503      0696      4      | If it's purgeable
: 504      0697      3      | WHILE traversing KFD's ordered KFE list
: 505      0698      2      | WHILE traversing KFD list
: 506      0699      2      END;
: 507      0700      2      RETURN TRUE;
: 508      0701      1      END;      ! Global routine PURGE
```

```

                                01FC 00000 PURGE: .WORD      Save R2,R3,R4,R5,R6,R7,R8
5E                                10 C2 00002      SUBL2      #16, SP
50 00000000G 00 D0 00005      MOVL      KFPB, R0
```

```

: 0598
:
: 0625
```


54		60	D0	0000C	MOVL	(R0), KFD	
		54	D5	0000F	1\$: TSTL	KFD	0632
		03	12	00011	BNEQ	2\$	
		0081	31	00013	BRW	7\$	
08	AE	0E	A4	9B	00016	2\$: MOVZBW	14(KFD), DEV_DSC
OC	AE	11	A4	9E	0001B	MOVAB	17(R4), DEV_DSC+4
	54		64	7D	00020	MOVQ	(KFD), KFD
			55	D5	00023	3\$: TSTL	KFE
			E8	13	00025	BEQL	1\$
56			55	D0	00027	MOVL	KFE, RET_KFE
55		04	A5	D0	0002A	MOVL	4(KFE), RFE
F1		11	A6	E8	0002E	BLBS	17(RET_KFE), 3\$
			7E	7C	00032	CLRQ	-(SP)
		08	AE	9F	00034	PUSHAB	CHAN
		14	AE	9F	00037	PUSHAB	DEV_DSC
00000000G	00		04	FB	0003A	CALLS	#4, SYSS\$ASSIGN
	50		6E	D0	00041	MOVL	CHAN, R0
		00000000G	00	16	00044	JSB	IOC\$VERIFYCHAN
	58		51	D0	0004A	MOVL	R1, R8
	12		02	DA	0004D	MTPR	#2, #18
		04	AE	9F	00050	PUSHAB	RET_KFD
			56	DD	00053	PUSHL	RET_KFE
0000V	CF		02	FB	00055	CALLS	#2, DETACH_KFE
	57		50	D0	0005A	MOVL	R0, STATUS
	13		57	E9	0005D	BLBC	STATUS, 4\$
		04	AE	D5	00060	TSTL	RET_KFD
			13	13	00063	BEQL	5\$
		04	AE	DD	00065	PUSHL	RET_KFD
0000V	CF		01	FB	00068	CALLS	#1, DEALLOC_PAGED
	57		50	D0	0006D	MOVL	R0, STATUS
	05		57	E8	00070	BLBS	STATUS, 5\$
	12		00	DA	00073	4\$: MTPR	#0, #18
			1B	11	00076	BRB	6\$
		0140	8F	BB	00078	5\$: PUSHR	#*M<R6, R8>
0000V	CF		02	FB	0007C	CALLS	#2, RETURN_KFE
	57		50	D0	00081	MOVL	R0, STATUS
	12		00	DA	00084	MTPR	#0, #18
			6E	DD	00087	PUSHL	CHAN
00000000G	00		01	FB	00089	CALLS	#1, SYSS\$DASSGN
	90		57	E8	00090	BLBS	STATUS, 3\$
	50		57	D0	00093	6\$: MOVL	STATUS, R0
				04	00096	RET	
	50		01	D0	00097	7\$: MOVL	#1, R0
			04	0009A	RET		

; Routine Size: 155 bytes, Routine Base: \$CODE\$ + 014D

; 509 0702 1

Page 16
(8)

```

511 0703 1 %SBTTL 'detach_kfe      Remove from Known File List';
512 0704 1
513 0705 1 ROUTINE DETACH_KFE (KFE, RET_KFD_ADR) =
514 0706 2 BEGIN
515 0707 2   +++
516 0708 2
517 0709 2   FUNCTIONAL DESCRIPTION:
518 0710 2
519 0711 2       Delete a Known File entry.
520 0712 2       If this was the last entry for the listhead, delete it
521 0713 2
522 0714 2   EXPLICIT INPUT:
523 0715 2
524 0716 2       none
525 0717 2
526 0718 2   ROUTINE VALUE:
527 0719 2
528 0720 2       R0 = return status, low bit set for success, else error status
529 0721 2
530 0722 2   ---
531 0723 2
532 0724 2 LOCAL
533 0725 2     CMPHSHKFE : REF BBLOCK,      ! KFE from hash table list to compare with
534 0726 2     CMPKFE : REF BBLOCK,        ! KFE from KFD to compare with
535 0727 2     HSHTAB : REF VECTOR[,LONG], ! Hash table
536 0728 2     PRVHSHKFE : REF BBLOCK,     ! Previous KFE from hash table
537 0729 2     PRVKFE : REF BBLOCK,        ! Previous KFE from KFD list
538 0730 2     STATUS;
539 0731 2
540 0732 2 MAP
541 0733 2     KFE : REF BBLOCK;
542 0734 2
543 0735 2 BIND
544 0736 2     KFPB = EXESGL KNOWN FILES : REF BBLOCK,
545 0737 2     RET_KFD = .RET_KFD_ADR;
546 0738 2
547 0739 2 IF .KFPB EQL 0
548 0740 2 THEN
549 0741 2     BEGIN
550 0742 2         INSSL INTRNLERR = EMPTYKFPB_ERR_DSC;
551 0743 2         RETURN INSSL_INTRNLERR;
552 0744 2     END;
553 0745 2
554 0746 2 HSHTAB = .KFPB [KFPB$KFEHSHTAB];
555 0747 2
556 0748 2 RET_KFD = 0;
557 0749 2
558 0750 2
559 0751 2 PRVHSHKFE = HSHTAB [.KFE [KFESB_HSHIDX]];
560 0752 2 CMPHSHKFE = .HSHTAB [.KFE [KFESB_HSHIDX]];
561 0753 2 WHILE .CMPHSHKFE NEQ 0 DO
562 0754 2     BEGIN
563 0755 2         IF .KFE EQL .CMPHSHKFE
564 0756 2         THEN
565 0757 2             BEGIN
566 0758 2                 LOCAL
567 0759 2                 KFD : REF BBLOCK;

```



```
: 568      0760      4
: 569      0761      4      PRVHSHKFE [KFESL_HSHLNK] = .KFE [KFESL_HSHLNK]; ! Remove it
: 570      0762      4      KFE [KFESL_HSHLNK] = 0;
: 571      0763      4      PRVHSHKFE = 0; ! Mark that it was found
: 572      0764      4      CMPHSHKFE = 0; ! Terminate traversal
: 573      0765      4
: 574      0766      4      KFD = .KFE [KFESL_KFD];
: 575      0767      4
: 576      0768      4      !
: 577      0769      4      ! Remove KFE from the ordered singly linked list starting in the
: 578      0770      4      ! KFD.
: 579      0771      4      !
: 580      0772      4      PRVKFE = .KFD;
: 581      0773      4      CMPKFE = .KFD [KFD$L_KFELIST];
: 582      0774      4
: 583      0775      4      WHILE .CMPKFE NEQ 0 DO
: 584      0776      5          BEGIN
: 585      0777      5              IF .CMPKFE EQL .KFE
: 586      0778      5                  THEN
: 587      0779      6                      BEGIN
: 588      0780      6                          PRVKFE [KFESL_KFELINK] = .KFE [KFESL_KFELINK]; ! Link around it.
: 589      0781      6                          KFE [KFESL_KFELINK] = 0; ! Clean up
: 590      0782      6                          PRVKFE = 0; ! mark successful traversal
: 591      0783      6                          CMPKFE = 0; ! terminate traversal
: 592      0784      6                      END
: 593      0785      5                  ELSE
: 594      0786      6                      BEGIN
: 595      0787      6                          PRVKFE = .CMPKFE;
: 596      0788      6                          CMPKFE = .CMPKFE [KFESL_KFELINK];
: 597      0789      5                      END;
: 598      0790      4                  END; ! While traversing ordered list of KFE's
: 599      0791      4
: 600      0792      4      !
: 601      0793      4      ! Traversed whole list and didn't find it, something is broken.
: 602      0794      4      !
: 603      0795      4      IF .PRVKFE NEQ 0
: 604      0796      4          THEN
: 605      0797      5              BEGIN
: 606      0798      5                  INSSL_INTRNLERR = KFDKFE_ERR_DSC;
: 607      0799      5                  RETURN INSSL_INTRNLERR;
: 608      0800      4              END;
: 609      0801      4
: 610      0802      4      KFD [KFD$W_REFCNT] = .KFD [KFD$W_REFCNT] - 1;
: 611      0803      4
: 612      0804      4      !
: 613      0805      4      ! If this was the only KFE off it's KFD, then disconnect the
: 614      0806      4      ! KFD.
: 615      0807      4      !
: 616      0808      4      IF .KFD [KFD$W_REFCNT] EQL 0
: 617      0809      4          THEN
: 618      0810      5              BEGIN
: 619      0811      5                  LOCAL
: 620      0812      5                      PRVKFD : REF BBLOCK,
: 621      0813      5                      CMPKFD : REF BBLOCK;
: 622      0814      5
: 623      0815      5                  RET KFD = .KFD;
: 624      0816      5                  PRVKFD = KFPB [KFPB$L_KFDLST];
```



```

625      0817 5      CMPKFD = .PRVKFD;
626      0818 5      WHILE .CMPKFD NEQ 0 DO
627      0819 6      BEGIN
628      0820 6      IF .KFD EQL .CMPKFD      ! This is it, disconnect it
629      0821 6      THEN
630      0822 7      BEGIN
631      0823 7      PRVKFD [KFD$L LINK] = .KFD [KFD$L LINK];
632      0824 7      KFD [KFD$L LINK] = 0;
633      0825 7      CMPKFD = 0;
634      0826 7      PRVKFD = 0;
635      0827 7      END
636      0828 6      ELSE
637      0829 7      BEGIN
638      0830 7      PRVKFD = .CMPKFD;
639      0831 7      CMPKFD = .CMPKFD [KFD$L LINK];
640      0832 6      END;
641      0833 5      END;      ! While traversing KFD list
642      0834 5
643      0835 5      IF .PRVKFD NEQ 0
644      0836 5      THEN
645      0837 6      BEGIN
646      0838 6      INSSL INTRNLERR = KFD_ERR_DSC;
647      0839 6      RETURN INSSL_INTRNLERR;
648      0840 5      END;
649      0841 5
650      0842 5      KFPB [KFPB$W_KFDLSTCNT] = .KFPB [KFPB$W_KFDLSTCNT] - 1;
651      0843 4      END;      ! Then remove KFD
652      0844 4      END
653      0845 3      ELSE      ! Same name, different file, keep traversing
654      0846 4      BEGIN
655      0847 4      PRVHSHKFE = .CMPHSHKFE;
656      0848 4      CMPHSHKFE = .CMPHSHKFE [KFES$L_HSHLNK];
657      0849 3      END;
658      0850 2      END;      ! WHILE traversing hash bucket list
659      0851 2
660      0852 2      !
661      0853 2      Have traversed whole list.
662      0854 2      !
663      0855 2      IF .PRVHSHKFE NEQ 0
664      0856 2      THEN
665      0857 3      BEGIN
666      0858 3      INSSL INTRNLERR = HSHKFE_ERR_DSC;
667      0859 3      RETURN INSSL_INTRNLERR;      ! It wasn't on the list it was supposed to be on
668      0860 2      END;
669      0861 2
670      0862 2      KFPB [KFPB$W_KFDLSTCNT] = .KFPB [KFPB$W_KFDLSTCNT] - 1;
671      0863 2
672      0864 2      RETURN TRUE;
673      0865 1      END;      ! routine DETACH_KFE

```

```

03FC 00000 DETACH_KFE:
59 00000000G 00 9E 00002      .WORD      Save R2,R3,R4,R5,R6,R7,R8,R9
                                MOVAB      INSSL_INTRNLERR, R9

```

: 0705
:

52	00000000G	00	D0	00009	MOVL	KFPB, R2	0739	
		07	12	00010	BNEQ	1\$	0742	
69	0000'	CF	9E	00012	MOVAB	EMPTYKFPB_ERR_DSC, INSSL_INTRNLERR	0743	
		5C	11	00017	BRB	7\$	0746	
53	04	A2	D0	00019	1\$: MOVL	4(R2), HSHTAB	0748	
	08	BC	D4	0001D	CLRL	@RET_KFD_ADR	0751	
51	04	AC	D0	00020	MOVL	KFE, R1	0752	
50	08	A1	9A	00024	MOVZBL	11(R1), R0	0753	
58	6340	DE	00028	MOVAL	(HSHTAB)[R0], PRVHSHKFE			
57	6340	D0	0002C	MOVL	(HSHTAB)[R0], CMPHSHKFE			
		57	D5	00030	2\$: TSTL	CMPHSHKFE		
		03	12	00032	BNEQ	3\$		
	0081	31	00034	BRW	15\$			
57		51	D1	00037	3\$: CMPL	R1, CMPHSHKFE		
		74	12	0003A	BNEQ	14\$		
68		61	D0	0003C	MOVL	(R1), (PRVHSHKFE)		
		61	D4	0003F	CLRL	(R1)		
		57	7C	00041	CLRQ	CMPHSHKFE		
50	0C	A1	D0	00043	MOVL	12(R1), KFD		
56		50	D0	00047	MOVL	KFD, PRVKFE		
55	04	A0	D0	0004A	MOVL	4(KFD), CMPKFE		
		1C	13	0004E	4\$: BEQL	6\$		
51		55	D1	00050	CMPL	CMPKFE, R1		
		0E	12	00053	BNEQ	5\$		
04	A6	04	A1	D0	00055	MOVL	4(R1), 4(PRVKFE)	
		04	A1	D4	0005A	CLRL	4(R1)	
		56	D4	0005D	CLRL	PRVKFE		
		55	D4	0005F	CLRL	CMPKFE		
		EB	11	00061	BRB	4\$		
56		55	D0	00063	5\$: MOVL	CMPKFE, PRVKFE		
55	04	A5	D0	00066	MOVL	4(CMPKFE), CMPKFE		
		E2	11	0006A	BRB	4\$		
		56	D5	0006C	6\$: TSTL	PRVKFE		
		07	13	0006E	BEQL	8\$		
69	0000'	CF	9E	00070	MOVAB	KFDKFE_ERR_DSC, INSSL_INTRNLERR		
		4A	11	00075	7\$: BRB	16\$		
	0C	A0	B7	00077	8\$: DECW	12(KFD)		
		B4	12	0007A	BNEQ	2\$		
08	BC	50	D0	0007C	MOVL	KFD, @RET_KFD_ADR		
54		52	D0	00080	MOVL	R2, PRVKFD		
53		54	D0	00083	MOVL	PRVKFD, CMPKFD		
		53	D5	00086	9\$: TSTL	CMPKFD		
		16	13	00088	BEQL	11\$		
53		50	D1	0008A	CMPL	KFD, CMPKFD		
		09	12	0008D	BNEQ	10\$		
64		60	D0	0008F	MOVL	(KFD), (PRVKFD)		
		60	D4	00092	CLRL	(KFD)		
		53	7C	00094	CLRQ	CMPKFD		
		EE	11	00096	BRB	9\$		
54		53	D0	00098	10\$: MOVL	CMPKFD, PRVKFD		
53		63	D0	0009B	MOVL	(CMPKFD), CMPKFD		
		E6	11	0009E	BRB	9\$		
		54	D5	000A0	11\$: TSTL	PRVKFD		
		07	13	000A2	BEQL	12\$		
69	0000'	CF	9E	000A4	MOVAB	KFD_ERR_DSC, INSSL_INTRNLERR		
		16	11	000A9	BRB	16\$		
	0C	A2	B7	000AB	12\$: DECW	12(R2)		

INSDELETE
V04-000

detach_kfe Remove from Known File List

L 2
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 20
(8)

58	80	11	000AE	13\$:	BRB	2\$	:	0755
57	57	D0	000B0	14\$:	MOVL	CMPHSHKFE, PRVSHKFE	:	0847
	67	D0	000B3		MOVL	(CMPHSHKFE), CMPHSHKFE	:	0848
	F6	11	000B6		BRB	13\$	:	0753
	58	D5	000B8	15\$:	TSTL	PRVSHKFE	:	0855
	0D	13	000BA		BEQL	17\$	:	
69	0000'	CF	9E	000BC	MOVAB	HSHKFE_ERR DSC, INSSL_INTRNLERR	:	0858
50	00000000G	8F	D0	000C1	MOVL	#INSS_INTRNLERR, R0	:	0859
		04	000C8		RET		:	
	0C	A2	B7	000C9	DECW	12(R2)	:	0862
50		01	D0	000CC	MOVL	#1, R0	:	0864
		04	000CF		RET		:	0865

; Routine Size: 208 bytes, Routine Base: \$CODE\$ + 01E8

; 674 0866 1

return_kfe Cleanup and deallocate entry

```
: 676      0867 1 %SBTTL 'return_kfe      Cleanup and deallocate entry';
: 677      0868 1
: 678      0869 1 ROUTINE RETURN_KFE (KFE,CCB) =
: 679      0870 2 BEGIN
: 680      0871 2 |+++
: 681      0872 2 |
: 682      0873 2 |   FUNCTIONAL DESCRIPTION:
: 683      0874 2 |
: 684      0875 2 |       Deallocate a Known File entry which has been removed from the list.
: 685      0876 2 |       If the file was installed /OPEN, take steps which will allow the
: 686      0877 2 |       file to be closed later.
: 687      0878 2 |
: 688      0879 2 |   EXPLICIT INPUT:
: 689      0880 2 |
: 690      0881 2 |       none
: 691      0882 2 |
: 692      0883 2 |   IMPLICIT INPUT:
: 693      0884 2 |
: 694      0885 2 |       KFE      Address of known file entry block to be returned
: 695      0886 2 |       CCB      Address of channel control block
: 696      0887 2 |
: 697      0888 2 |   ROUTINE VALUE:
: 698      0889 2 |
: 699      0890 2 |       R0 = return status, low bit set for success, else error status
: 700      0891 2 |
: 701      0892 2 |   ---
: 702      0893 2 |
: 703      0894 2 |   LOCAL
: 704      0895 2 |       PTR,
: 705      0896 2 |       WCB : REF BBLOCK,
: 706      0897 2 |       STATUS;
: 707      0898 2 |
: 708      0899 2 |   MAP
: 709      0900 2 |       CCB : REF $BBLOCK,
: 710      0901 2 |       KFE : REF BBLOCK;
: 711      0902 2 |
: 712      0903 2 |   BIND
: 713      0904 2 |       SUFFIX = UPLIT (%ASCII '_001') : VECTOR [,BYTE];
: 714      0905 2 |
: 715      0906 2 |   IF .KFE [KFES$V_HDRRES]
: 716      0907 2 |   THEN
: 717      0908 3 |       BEGIN
: 718      0909 3 |       BIND
: 719      0910 3 |           KFRH = .KFE [KFES$L_IMGHDR] - KFRH$C_LENGTH;
: 720      0911 3 |
: 721      0912 3 |       IF .KFE [KFES$L_IMGHDR] EQL 0
: 722      0913 3 |       THEN
: 723      0914 4 |           BEGIN
: 724      0915 4 |               INSSL INTRNLERR = IMGHDR_ERR_DSC;
: 725      0916 4 |               RETURN INSSL_INTRNLERR;
: 726      0917 3 |           END;
: 727      0918 3 |
: 728      0919 3 |       STATUS = DEALLOC_PAGED (KFRH);
: 729      0920 3 |       END;
: 730      0921 2 |
: 731      0922 2 |   !
: 732      0923 2 |   ! Delete any global sections created if this was installed /SHARE
```

```

733 0924 2 !
734 0925 2 IF .KFE [KFESV_SHARED]
735 0926 2 THEN
736 0927 2 BEGIN
737 0928 2 LOCAL
738 0929 2     GBLSECNAM_DSC : BBLOCK [DSC$C S BLN],
739 0930 2     GBLSECNAM : BBLOCK [INS$C_GBLNAMLEN];
740 0931 2
741 0932 2     !
742 0933 2     Initialize
743 0934 2     !
744 0935 2     GBLSECNAM_DSC = 0;
745 0936 2     GBLSECNAM_DSC [DSC$A POINTER] = GBLSECNAM;
746 0937 2     CH$FILL (0, INS$C_GBLNAMLEN, GBLSECNAM);
747 0938 2     GBLSECNAM_DSC [DSC$W_LENGTH] = .KFE [KFESB_FILNAMLEN] + 4;
748 0939 2     PTR = CH$MOVE(.KFE [KFESB_FILNAMLEN], KFE [KFEST_FILNAM], GBLSECNAM);
749 0940 2     CH$MOVE(4, SUFFIX, .PTR);
750 0941 2
751 0942 2     !
752 0943 2     At this point, we convert the match control to be 'MATEQU'. This
753 0944 2     ensures that we only delete global sections that are likely to
754 0945 2     have been created at the time this image was installed. This avoids
755 0946 2     the problem where a newer copy of the global section set with, most
756 0947 2     likely, a greater minor ID from being deleted instead if the match
757 0948 2     control in the KFE happened to be 'MATLEQ', as is typical.
758 0949 2
759 0950 2     For instance, if the shareable image is installed twice (with
760 0951 2     different KFDs, of course) and the global section ident for the
761 0952 2     second image has a larger minor ID than the first, then two
762 0953 2     different sets of global sections will be created. Now, if the
763 0954 2     match control is set, as is typical, to be 'MATLEQ', this means
764 0955 2     that not only will the second set of global sections be referenced
765 0956 2     for both images during image activation, but should either image
766 0957 2     be deleted, the second set of global section will also be deleted!
767 0958 2     This is incorrect in the case of the first image being deleted because
768 0959 2     the first set of global sections had been created just for that image
769 0960 2     and the second image may not be able to work with the first set of
770 0961 2     global sections.
771 0962 2
772 0963 2     To make a long story short, by converting the match control to 'MATEQU',
773 0964 2     we ensure deletion of the correct set of global sections.
774 0965 2
775 0966 2     KFE [KFESB_MATCHCTL] = ISD$K_MATEQU;
776 0967 2
777 0968 2     !
778 0969 2     Delete all the global sections
779 0970 2
780 0971 2     INCR I FROM 1 TO .KFE [KFESW_GBLSECCNT] DO
781 0972 2     BEGIN
782 0973 2     STATUS = $DGBLSC (
783 0974 2     FLAGS = SEC$M_SYSGBL,
784 0975 2     GSDNAM = GBLSECNAM_DSC,
785 0976 2     IDENT = KFE [KFESB_MATCHCTL]
786 0977 2     );
787 0978 2     INS$BLD_GBLSECNAM (GBLSECNAM_DSC);
788 0979 2     END;
789 0980 2

```

! Deleting a system global section
! Global section name
! Address of match control and ident quadword
! For as many global sections as were created

return_kfe Cleanup and deallocate entry

```

790 0981 2      END;                ! Delete global sections
791 0982 2
792 0983 2      IF .KFE [KFESV_OPEN]
793 0984 2      THEN
794 0985 2          IF .CCB EQL 0
795 0986 2          THEN
796 0987 2              BEGIN
797 0988 2                  WCB = .KFE [KFESL_WCB];
798 0989 2                  WCB [WCB$W_REFCNT] = .WCB [WCB$W_REFCNT] - 1;    ! Allow file to be closed
799 0990 2                  END
800 0991 2      ELSE
801 0992 2          !
802 0993 2          ! Jam the WCB address into the CCB so that the file can be closed later.
803 0994 2          !
804 0995 2          CCB [CCB$W_WIND] = .KFE [KFESL_WCB];
805 0996 2
806 0997 2
807 0998 2      IF .IN$GL_CTLMSK [IN$V_REPLACE]
808 0999 2      THEN
809 1000 2          !
810 1001 2          ! Preserve attributes so they can be duplicated on a REPLACE
811 1002 2          !
812 1003 2          BEGIN
813 1004 2              IN$GL_REPLACE_MSK [IN$V_OPEN] = .KFE [KFESV_OPEN];
814 1005 2              IN$GL_REPLACE_MSK [IN$V_SHARED] = .KFE [KFESV_SHARED];
815 1006 2              IN$GL_REPLACE_MSK [IN$V_PRIV] = .KFE [KFESV_PROCPRIV];
816 1007 2              CH$MOVE (8, KFE [KFESQ_PROCPRIV], IN$GQ_KFEPRIVS);
817 1008 2              IN$GL_REPLACE_MSK [IN$V_NOPURGE] = .KFE [KFESV_NOPURGE];
818 1009 2              IN$GL_REPLACE_MSK [IN$V_ACCOUNT] = .KFE [KFESV_ACCOUNT];
819 1010 2              IN$GL_REPLACE_MSK [IN$V_HDRRES] = .KFE [KFESV_HDRRES];
820 1011 2              IN$GL_REPLACE_MSK [IN$V_PROTECT] = .KFE [KFESV_PROTECT];
821 1012 2              IN$GL_REPLACE_MSK [IN$V_WRITABLE] = .KFE [KFESV_WRITEABLE];
822 1013 2              IN$GL_REPLACE_MSK [IN$V_EXEONLY] = .KFE [KFESV_EXEONLY];
823 1014 2          END;
824 1015 2
825 1016 2      STATUS = DEALLOC_PAGED (.KFE);
826 1017 2
827 1018 2      RETURN TRUE;
828 1019 1      END;                ! routine RETURN_KFE
```

.PSECT \$SPLITS,NOWRT,NOEXE,2

31 30 30 5F 000D8 P.AAK: .ASCII _001\ ;

SUFFIX=

P.AAK

.EXTRN SYS\$DGBLSC

.PSECT \$CODE\$,NOWRT,2

03FC 00000 RETURN_KFE:

```

59 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9
5E          34 C2 00009 MOVAB IN$GL_REPLACE_MSK, R9
56          04 AC D0 0000C SUBL2 #52, SP
57          10 A6 9E 00010 MOVL KFE, R6
                    MOVAB 16(R6), R7
```

```

: 0869
:
: 0906
:
```

INSDELETE
V04-000

return_kfe Cleanup and deallocate entry

C 3
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 24
(9)

25		67		04	E1	00014	BBC	#4, (R7), 2\$		
50	1C	A6		0C	C3	00018	SUBL3	#12, 28(R6), R0	0910	
			1C	A6	D5	0001D	TSTL	28(R6)	0912	
				11	12	00020	BNEQ	1\$		
	00000000G	00	0000'	CF	9E	00022	MOVAB	IMGHDR_ERR_DSC, INSSL_INTRNLERR	0915	
		50	00000000G	8F	D0	0002B	MOVL	#INSSL_INTRNLERR, R0	0916	
					04	00032	RET			
				50	DD	00033	PUSHL	R0	0919	
	0000V	CF		01	FB	00035	CALLS	#1, DEALLOC_PAGED		
		58		50	D0	0003A	MOVL	R0, STATUS		
53		67		05	E1	0003D	BBC	#5, (R7), 5\$	0925	
			2C	AE	D4	00041	CLRL	GBLSECNAM_DSC	0935	
	30	AE		6E	9E	00044	MOVAB	GBLSECNAM, GBLSECNAM_DSC+4	0936	
2B	00	6E		00	2C	00048	MOVCS	#0, (SP), #0, #43, GBLSECNAM	0937	
				6E		0004D				
	2C	AE	36	A6	9B	0004E	MOVZBW	54(R6), GBLSECNAM_DSC	0938	
	2C	AE		04	A0	00053	ADDW2	#4, GBLSECNAM_DSC		
		50	36	A6	9A	00057	MOVZBL	54(R6), R0	0939	
6E	37	A6		50	28	0005B	MOVCS	R0, 55(R6), GBLSECNAM		
		63	0000'	CF	D0	00060	MOVL	SUFFIX, (PTR)	0940	
	28	A6		01	90	00065	MOVB	#1, 40(R6)	0966	
		53	12	A6	3C	00069	MOVZWL	18(R6), R3	0971	
				52	D4	0006D	CLRL	I		
				1F	11	0006F	BRB	4\$		
			28	A6	9F	00071	PUSHAB	40(R6)	0977	
			30	AE	9F	00074	PUSHAB	GBLSECNAM_DSC		
			8000	8F	3C	00077	MOVZWL	#32768, -7SP)		
	00000000G	00		03	FB	0007C	CALLS	#3, SYSSDGBLSC		
		58		50	D0	00083	MOVL	R0, STATUS		
			2C	AE	9F	00086	PUSHAB	GBLSECNAM_DSC	0978	
	00000000G	00		01	FB	00089	CALLS	#1, INSSLB[0]_GBLSECNAM		
DD		52		53	F3	00090	AOBLEQ	R3, I, 3\$	0971	
14		67		03	E1	00094	BBC	#3, (R7), 7\$	0983	
		51	08	AC	D0	00098	MOVL	CCB, R1	0985	
				09	12	0009C	BNEQ	6\$		
		50	18	A6	D0	0009E	MOVL	24(R6), WCB	0988	
			0E	A0	B7	000A2	DECW	14(WCB)	0989	
				05	11	000A5	BRB	7\$	0985	
			18	A6	D0	000A7	MOVL	24(R6), 4(R1)	0995	
	04	A1		05	E1	000AC	BBC	#5, INSSLGL_CTLMSK, 8\$	0998	
63	00000000G	00		03	EF	000B4	EXTZV	#3, #1, (R7), R0	1004	
01		01		50	F0	000B9	INSV	R0, #5, #1, INSSLGL_REPLACE_MSK+1		
50		67		05	EF	000BF	EXTZV	#5, #1, (R7), R0	1005	
02		01		50	F0	000C4	INSV	R0, #1, #1, INSSLGL_REPLACE_MSK+2		
50		67		02	EF	000CA	EXTZV	#2, #1, (R7), R0	1006	
01		01		50	F0	000CF	INSV	R0, #7, #1, INSSLGL_REPLACE_MSK+1		
00000000G		00		08	28	000D5	MOVCS	#8, 32(R6), INSSLGL_KFEPRIVS	1007	
02		01	20	A6	A7	000DE	INSV	1(R7), #3, #1, INSSLGL_REPLACE_MSK+2	1008	
50		67		01	09	000E5	EXTZV	#9, #1, (R7), R0	1009	
02		01		50	F0	000EA	INSV	R0, #4, #1, INSSLGL_REPLACE_MSK+2		
50		67		04	EF	000F0	EXTZV	#4, #1, (R7), R0	1010	
01		01		50	F0	000F5	INSV	R0, #6, #1, INSSLGL_REPLACE_MSK+1		
02		01		67	F0	000FB	INSV	(R7), #0, #1, INSSLGL_REPLACE_MSK+2	1011	
50		67		0A	EF	00101	EXTZV	#10, #1, (R7), R0	1012	
02		01		50	F0	00106	INSV	R0, #2, #1, INSSLGL_REPLACE_MSK+2		
50		67		0B	EF	0010C	EXTZV	#11, #1, (R7), R0	1013	
02		01		50	F0	00111	INSV	R0, #5, #1, INSSLGL_REPLACE_MSK+2		

INSDELETE
V04-000

return_kfe Cleanup and deallocate entry

D 3
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 25
(9)

0000V	CF	56	DD	00117	8\$:	PUSHL	R6	
	58	01	FB	00119		CALLS	#1,	DEALLOC_PAGED
	50	50	DO	0011E		MOVL	R0,	STATUS
	50	01	DO	00121		MOVL	#1,	R0
		04	00124			RET		

: 1016
:
:
:
: 1018
:
: 1019

; Routine Size: 293 bytes, Routine Base: \$CODE\$ + 02B8

; 829 1020 1

```
1021 1 %SBTTL 'INSS$FIND_KFE Locate KFE given name and hash bucket';
1022 1
1023 1 GLOBAL ROUTINE INSS$FIND_KFE (HSHIDX, NAM) =
1024 2 BEGIN
1025 2 !+++
1026 2
1027 2 FUNCTIONAL DESCRIPTION:
1028 2
1029 2 This routine locates a KFE (if one exists) for the specified
1030 2 filespec.
1031 2
1032 2 EXPLICIT INPUT:
1033 2
1034 2 none
1035 2
1036 2 ROUTINE VALUE:
1037 2
1038 2 R0 = 0 if no such known file exists
1039 2 else, KFE address
1040 2
1041 2 ---
1042 2
1043 2 LOCAL
1044 2 CMPKFE : REF BBLOCK,
1045 2 HSHTAB : REF VECTOR [,LONG],
1046 2 PRVKFE : REF BBLOCK,
1047 2 STATUS;
1048 2
1049 2 MAP
1050 2 NAM : REF BBLOCK;
1051 2
1052 2 BIND
1053 2 KFPB = EXE$GL_KNOWN_FILES : REF BBLOCK;
1054 2
1055 2 IF .KFPB EQL 0 THEN RETURN 0;
1056 2
1057 2 HSHTAB = .KFPB [KFPB$L_KFEHSHTAB];
1058 2 IF .HSHTAB [.HSHIDX] EQL 0 THEN RETURN 0;
1059 2
1060 2 PRVKFE = HSHTAB [.HSHIDX]; ! Previous KFE
1061 2 CMPKFE = .HSHTAB [.HSHIDX]; ! Comparison KFE
1062 2 WHILE .CMPKFE NEQ 0 DO ! Single linked list ending in zero
1063 3 BEGIN
1064 3 CASE CH$COMPARE (.NAM [NAM$B_NAME], .NAM [NAM$L_NAME],
1065 3 .CMPKFE [KFE$B_FILNAMLEN], CMPKFE [KFE$B_FILNAM], %C' ')
1066 3 FROM -1 TO 1 OF ! Either less than, equal to, or greater than
1067 3 SET
1068 3
1069 3 [-1]: ! Less than, therefore its not in the list, ERROR
1070 4 BEGIN
1071 4 RETURN 0;
1072 4 END;
1073 3
1074 3 [0] : ! Same file name, check if KFD is the same
1075 4 BEGIN
1076 4 LOCAL
1077 4 DDTSTR : BBLOCK [NAM$C_MAXRSS],
```



```

888      1078 4      DDT_DSC : $BBLOCK [DSC$C_S_BLN],
889      1079 4      KFD : REF BBLOCK;
890      1080 4
891      1081 4      KFD = .CMPKFE [KFES$L_KFD];
892      1082 4
893      1083 4      Build an ASCII string of the concatenated Device, Directory
894      1084 4      Type strings.
895      1085 4
896      1086 4      DDT_DSC [DSC$W_LENGTH] = .NAM [NAM$B_DEV] + .NAM [NAM$B_DIR] +
897      1087 4      .NAM [NAM$B_TYPE];      ! Length of DDT string
898      1088 4
899      1089 4      DDT_DSC [DSC$A_POINTER] = DDTSTR;      ! DDT string address
900      1090 4
901      1091 4      DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAM [NAM$B_DEV], .NAM [NAM$B_DIR],
902      1092 4      .DDT_DSC [DSC$A_POINTER]);
903      1093 4      DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAM [NAM$B_DIR], .NAM [NAM$B_TYPE],
904      1094 4      .DDT_DSC [DSC$A_POINTER]);
905      1095 4      DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAM [NAM$B_TYPE], .NAM [NAM$B_DEV],
906      1096 4      .DDT_DSC [DSC$A_POINTER]);
907      1097 4
908      1098 4      DDT_DSC [DSC$A_POINTER] = DDTSTR;      ! DDT string address
909      1099 4
910      1100 4      INSS$CVT_DIR (DDT_DSC);      ! Convert and compress brackets
911      1101 4
912      1102 4      IF CH$COMPARE (.DDT_DSC [DSC$W_LENGTH], DDTSTR,
913      1103 4      .KFD [KFD$B_DDTSTRLEN], KFD [KFD$T_DDTSTR], %C' ')
914      1104 4      EQL 0
915      1105 4      THEN
916      1106 4      RETURN .CMPKFE;      ! They were the same
917      1107 4
918      1108 4      Same file name, different file, keep traversing
919      1109 4
920      1110 4      PRVKFE = .CMPKFE;
921      1111 4      CMPKFE = .CMPKFE [KFES$L_HSHLNK];
922      1112 4      END;
923      1113 3
924      1114 3      [1] : ! Greater than, keep looking
925      1115 3      BEGIN
926      1116 4      PRVKFE = .CMPKFE;
927      1117 4      CMPKFE = .CMPKFE [KFES$L_HSHLNK];
928      1118 4      END;
929      1119 3      TES;
930      1120 3      END;      ! WHILE traversing hash bucket list
931      1121 2
932      1122 2      Have traversed whole list and didn't find it.
933      1123 2
934      1124 2      RETURN 0;
935      1125 2
936      1126 2
937      1127 2
938      1128 1      END;      ! routine INSS$FIND_KFE

```

INSDELETE
V04-000

INSS\$FIND_KFE Locate KFE given name and hash buc

G 3
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 28
(10)

5E	FEF8	CE	9E	00002	MOVAB	-264(SP), SP	
50	00000000G	00	D0	00007	MOVL	KFPB, R0	1055
		1B	13	0000E	BEQL	2\$	
51	04	A0	D0	00010	MOVL	4(R0), HSHTAB	1057
50	04	AC	D0	00014	MOVL	HSHTAB, R0	1058
		6140	D5	00018	TSTL	(HSHTAB)[R0]	
		0E	13	0001B	BEQL	2\$	
59		6140	DE	0001D	MOVAL	(HSHTAB)[R0], PRVKFE	1060
57		6140	D0	00021	MOVL	(HSHTAB)[R0], CMPKFE	1061
56	08	AC	D0	00025	MOVL	NAM, R6	1064
		57	D5	00029	TSTL	CMPKFE	1062
		03	12	0002B	BNEQ	3\$	
		008D	31	0002D	BRW	6\$	
51	3B	A6	9A	00030	MOVZBL	59(R6), R1	1064
50	36	A7	9A	00034	MOVZBL	54(CMPKFE), R0	1065
B6		51	2D	00038	CMPC5	R1, @76(R6), #32, R0, 55(CMPKFE)	
	37	A7		0003E			
		72	1A	00040	BGTRU	5\$	
		79	1F	00042	BLSSU	6\$	
58	0C	A7	D0	00044	MOVL	12(CMPKFE), KFD	1081
50	39	A6	9A	00048	MOVZBL	57(R6), R0	1086
51	3A	A6	9A	0004C	MOVZBL	58(R6), R1	
50		51	C0	00050	ADDL2	R1, R0	
52	3C	A6	9A	00053	MOVZBL	60(R6), R2	1087
50		52	A1	00057	ADDW3	R2, R0, DDT_DSC	
6E		AE	9E	0005B	MOVAB	DDTSTR, DDT_DSC+4	1089
	04	AE	9A	00060	MOVZBL	57(R6), R0	1091
04	BE	44	B6	00064	MOVZBL	R0, @68(R6), @DDT_DSC+4	1092
		04	AE	0006A	MOVZBL	R3, DDT_DSC+4	
50		3A	A6	0006E	MOVZBL	58(R6), R0	1093
04	BE	48	B6	00072	MOVZBL	R0, @72(R6), @DDT_DSC+4	1094
		04	AE	00078	MOVZBL	R3, DDT_DSC+4	
50		3C	A6	0007C	MOVZBL	60(R6), R0	1095
04	BE	50	B6	00080	MOVZBL	R0, @80(R6), @DDT_DSC+4	1096
		04	AE	00086	MOVZBL	R3, DDT_DSC+4	
		04	AE	0008A	MOVZBL	DDTSTR, DDT_DSC+4	1098
		08	AE	0008F	PUSHL	SP	1100
	00000000G	00	01	00091	CALLS	#1, INSS\$CVT_DIR	
		50	A8	00098	MOVZBL	16(KFD), R0	1103
		54	01	0009C	MOVZBL	#1, R4	
50	20	08	AE	0009F	CMPC5	DDT_DSC, DDTSTR, #32, R0, 17(KFD)	
		11	A8	000A5			
		03	1A	000A7	BGTRU	4\$	
54		01	D9	000A9	SBWC	#1, R4	
		54	D5	000AC	TSTL	R4	1104
		04	12	000AE	BNEQ	5\$	
50		57	D0	000B0	MOVL	CMPKFE, R0	1106
		04	000B3		RET		
59		57	D0	000B4	MOVL	CMPKFE, PRVKFE	1117
57		67	D0	000B7	MOVL	(CMPKFE), CMPKFE	1118
		FF6C	31	000BA	BRW	1\$	1062
		50	D4	000BD	CLRL	R0	1128
		04	000BF		RET		

; Routine Size: 192 bytes, Routine Base: \$CODE\$ + 03DD

INSDELETE
V04-000

: 939

INS\$FIND_KFE Locate KFE given name and hash buc

1129 1

H 3
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 29
(10)

```
; 941      1130 1 %SBTTL 'dealloc_paged Return memory to paged pool';
; 942      1131 1
; 943      1132 1 ROUTINE DEALLOC_PAGED (ADR) =
; 944      1133 2 BEGIN
; 945      1134 2 !+++
; 946      1135 2
; 947      1136 2 FUNCTIONAL DESCRIPTION:
; 948      1137 2
; 949      1138 2 Deallocate the paged pool which is specified by ADR.
; 950      1139 2 Size of block to return is specified in size offset
; 951      1140 2 start of block.
; 952      1141 2
; 953      1142 2 !---
; 954      1143 2
; 955      1144 2 LOCAL
; 956      1145 2 STATUS;
; 957      1146 2
; 958      1147 2 STATUS = EXE$DEAPAGED (.ADR); ! Deallocate to paged pool
; 959      1148 2
; 960      1149 2 RETURN .STATUS;
; 961      1150 1 END; ! Routine DEALLOC_PAGED
```

```
OFFC 00000 DEALLOC_PAGED:
50      04 AC D0 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 ; 1132
00000000G 00 16 00006 MOVL ADR, R0 ; 1147
04 0000C JSB EXE$DEAPAGED ; 1150
RET
```

; Routine Size: 13 bytes, Routine Base: \$CODE\$ + 049D

; 962 1151 1

INSDELETE
V04-000

dealloc_paged Return memory to paged pool

J 3
16-Sep-1984 01:52:50
14-Sep-1984 12:35:37

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSDELETE.B32;1

Page 31
(12)

: 964 1152 1 END ! Module insdelete
: 965 1153 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	220	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
\$CODES	1194	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	61	0	1000	00:01.8

COMMAND QUALIFIERS

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

: Size: 1194 code + 220 data bytes
: Run Time: 00:25.3
: Elapsed Time: 01:21.1
: Lines/CPU Min: 2738
: Lexemes/CPU-Min: 17337
: Memory Used: 167 pages
: Compilation Complete

0189

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY