


```

FFFFFFFFF  IIIIII  NN      NN  DDDDDDD  HH      HH  EEEEEEEEE  LL      DDDDDDD
FFFFFFFFF  IIIIII  NN      NN  DDDDDDD  HH      HH  EEEEEEEEE  LL      DDDDDDD
FF          II      NN      NN  DD        DD  HH      HH  EE          LL      DD        DD
FF          II      NN      NN  DD        DD  HH      HH  EE          LL      DD        DD
FF          II      NNNN    NN  DD        DD  HH      HH  EE          LL      DD        DD
FF          II      NNNN    NN  DD        DD  HH      HH  EE          LL      DD        DD
FFFFFFFFF  II      NN  NN  NN  DD        DD  HHHHHHHHH  EEEEEEE    LL      DD        DD
FFFFFFFFF  II      NN  NN  NN  DD        DD  HHHHHHHHH  EEEEEEE    LL      DD        DD
FF          II      NN      NNNN  DD        DD  HH      HH  EE          LL      DD        DD
FF          II      NN      NNNN  DD        DD  HH      HH  EE          LL      DD        DD
FF          II      NN      NN  DD        DD  HH      HH  EE          LL      DD        DD
FF          II      NN      NN  DD        DD  HH      HH  EE          LL      DD        DD
FF          IIIIII  NN      NN  DDDDDDD  HH      HH  EEEEEEEEE  LLLLLLLLL  DDDDDDD
FF          IIIIII  NN      NN  DDDDDDD  HH      HH  EEEEEEEEE  LLLLLLLLL  DDDDDDD

```

```

LL          IIIIII  SSSSSSS
LL          IIIIII  SSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSS
LL          II      SSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          IIIIII  SSSSSSS
LLLLLLLLLL IIIIII  SSSSSSS
LLLLLLLLLL IIIIII  SSSSSSS

```

```
1 0001 0 %TITLE 'FINDHELD - SYSS$FIND_HELD loadable system service'
2 0002 0 MODULE FINDHELD (IDENT = 'V03-000') =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY: EXECUTIVE, SYSTEM SERVICES
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains the loadable system service SYSS$FIND_HELD.
36 0036 1
37 0037 1 ENVIRONMENT:
38 0038 1
39 0039 1 VAX/VMS native mode, user, supervisor, or exec modes.
40 0040 1
41 0041 1 --
42 0042 1
43 0043 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 16-Nov-1982 18:51
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1 V03-004 TMK0001 Todd M. Katz 22-Oct-1983
48 0048 1 The name of the real system service entry point for
49 0049 1 $FINISH_RDB has been changed from EXE$FINISH_RDB to
50 0050 1 EXE$$FINISH_RDB. This change was required because the
51 0051 1 system service could no longer be reached directly from the
52 0052 1 executive mode system service dispatcher.
53 0053 1
54 0054 1 V03-003 RSH0039 R. Scott Hanna 21-Jun-1983
55 0055 1 Changes to open the rights database as a process
56 0056 1 permanent file when there is no image active.
57 0057 1
```

```
58 0058 1 | V03-002 GAS0126 Gerry Smith 26-May-1983
59 0059 1 | Put SYSS$FIND_HELD into its own module. This is
60 0060 1 | necessary so that LOGINOUT can reference it before
61 0061 1 | the loadable system services are loaded, at boot time.
62 0062 1 |
63 0063 1 | V03-001 RSH0008 R. Scott Hanna 01-Mar-1983
64 0064 1 | Changed SYSS$CREATE_RDB to call EXES$SET_RDIPTR.
65 0065 1 |
66 0066 1 | **
67 0067 1 |
68 0068 1 |
69 0069 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
70 0070 1 |
71 0071 1 | FORWARD ROUTINE
72 0072 1 | SYSS$FIND_HELD;
73 0073 1 |
74 0074 1 | EXTERNAL ROUTINE
75 0075 1 | EXES$OPEN_RDB : ADDRESSING_MODE (ABSOLUTE),
76 0076 1 | EXES$CLOSE_RDB : NOVALUE ADDRESSING_MODE (ABSOLUTE),
77 0077 1 | EXES$FINISH_RDB : ADDRESSING_MODE (ABSOLUTE);
78 0078 1 |
79 0079 1 | BUILTIN
80 0080 1 | PROBER,
81 0081 1 | PROBEW;
82 0082 1 |
```

```

84 0083 1 GLOBAL ROUTINE SYSS$FIND_HELD (HOLDER, ID, ATTRIB, CONTXT) =
85 0084 1
86 0085 1 ++
87 0086 1
88 0087 1 FUNCTIONAL DESCRIPTION:
89 0088 1
90 0089 1 This routine searches the rights database for all identifiers
91 0090 1 held by the specified holder, and returns their identifier and
92 0091 1 attributes.
93 0092 1
94 0093 1 CALLING SEQUENCE:
95 0094 1 SYSS$FIND_HELD (HOLDER, ID, ATTRIB, CONTXT)
96 0095 1
97 0096 1 INPUT PARAMETERS:
98 0097 1 HOLDER: address of a quadword which specifies the holder
99 0098 1 id of the records to find
100 0099 1 CONTXT: (optional) address of a longword containing the
101 0100 1 record stream context. initially should be zero,
102 0101 1 value output first call, value input on subsequent
103 0102 1 calls.
104 0103 1
105 0104 1 IMPLICIT INPUTS:
106 0105 1 NONE
107 0106 1
108 0107 1 OUTPUT PARAMETERS:
109 0108 1 ID: (optional) address to return the identifier longword
110 0109 1 ATTRIB: (optional) address to return the holder attributes longword
111 0110 1
112 0111 1 IMPLICIT OUTPUTS:
113 0112 1 NONE
114 0113 1
115 0114 1 ROUTINE VALUE:
116 0115 1 Status of operation
117 0116 1
118 0117 1 SIDE EFFECTS:
119 0118 1 NONE
120 0119 1
121 0120 1 --
122 0121 1
123 0122 2 BEGIN
124 0123 2
125 0124 2 LOCAL
126 0125 2 LOC HOLDER : REF VECTOR, ! local copy of HOLDER
127 0126 2 HOLDER_ID : VECTOR [2], ! local copy of holder id quadword
128 0127 2 LOC_ID : LONG, ! local copy of ID
129 0128 2 LOC_ATTRIB : LONG, ! local copy of ATTRIB
130 0129 2 LOC_CONTXT : LONG, ! local copy of CONTXT
131 0130 2 STATUS : LONG, ! general status value
132 0131 2 CONTINUE : LONG, ! flag indicating continuation
133 0132 2 CLOSE : LONG, ! call EXESCLOSE RDB flag
134 0133 2 RAB : $RAB DECL, ! RAB for file I70
135 0134 2 REC_BUFFER : $BBLOCK [KGB$K_HOLD_RECORD];
136 0135 2 ! record buffer to read records
137 0136 2
138 0137 2 LABEL
139 0138 2 RDB_OPEN; ! rights database is open in this block
140 0139 2

```

```

141 0140 2 ! Validate parameters
142 0141 2 !
143 0142 2 !
144 0143 2 LOC HOLDER = .HOLDER;
145 0144 2 IF NOT PROBER (%REF(0), %REF(8), .LOC_HOLDER) THEN RETURN SSS_ACCVIO;
146 0145 2 HOLDER_ID[0] = .LOC_HOLDER[0];
147 0146 2 HOLDER_ID[1] = .LOC_HOLDER[1];
148 0147 2 IF .HOLDER_ID[0] GTRU UIC$K_MAX_UIC OR .HOLDER_ID[1] NEQU 0
149 0148 2 THEN
150 0149 2 RETURN SSS_IVIDENT;
151 0150 2 !
152 0151 2 LOC_ID = .ID;
153 0152 2 IF .LOC_ID NEQU 0 AND NOT PROBEW (%REF(0), %REF(4), .LOC_ID)
154 0153 2 THEN
155 0154 2 RETURN SSS_ACCVIO;
156 0155 2 !
157 0156 2 LOC_ATTRIB = .ATTRIB;
158 0157 2 IF .LOC_ATTRIB NEQU 0 AND NOT PROBEW (%REF(0), %REF(4), .LOC_ATTRIB)
159 0158 2 THEN
160 0159 2 RETURN SSS_ACCVIO;
161 0160 2 !
162 0161 2 LOC_CONTXT = .CONTXT;
163 0162 2 IF .LOC_CONTXT NEQU 0 AND NOT PROBEW (%REF(0), %REF(4), .LOC_CONTXT)
164 0163 2 THEN
165 0164 2 RETURN SSS_ACCVIO;
166 0165 2 !
167 0166 2 ! Open the rights database for reading. Record whether this is an initial
168 0167 2 ! call or a continuation by checking if the context is zero or not.
169 0168 2 !
170 0169 2 !
171 0170 2 CONTINUE = (IF .LOC_CONTXT NEQU 0 THEN ..LOC_CONTXT NEQU 0 ELSE 0);
172 0171 2 !
173 P 0172 2 $RAB_INIT (RAB = RAB,
174 P 0173 2 RAC = KEY,
175 P 0174 2 KRF = 1,
176 P 0175 2 KSZ = KGB$S HOLDER,
177 P 0176 2 KBF = HOLDER_ID[0],
178 P 0177 2 ROP = (WAT, NLK, LIM),
179 P 0178 2 USZ = KGB$K HOLD_RECORD,
180 P 0179 2 UBF = REC_BUFFER
181 0180 2 );
182 0181 2 STATUS = EXE$OPEN RDB (.LOC_CONTXT, 0, RAB[RAB$W_ISI], CLOSE);
183 0182 2 IF NOT .STATUS THEN RETURN .STATUS;
184 0183 2 !
185 0184 2 RDB_OPEN:
186 0185 2 _BEGIN
187 0186 2 !
188 0187 2 ! On an initial call, do an indexed $GET to position to the first
189 0188 2 ! holder record. Otherwise we do sequential gets.
190 0189 2 !
191 0190 2 !
192 0191 2 IF NOT .CONTINUE
193 0192 2 THEN
194 0193 2 BEGIN
195 0194 2 STATUS = $GET (RAB = RAB);
196 0195 2 IF .STATUS EQLU RMSS_RNF THEN STATUS = SSS_NOSUCHID;
197 0196 2 IF NOT .STATUS

```

```

198 0197 4 THEN
199 0198 3 BEGIN
200 0199 5 EXESS$FINISH_RDB (.LOC_CONTXT);
201 0200 5 LEAVE RDB_OPEN;
202 0201 4 END;
203 0202 4 END
204 0203 3 ELSE
205 0204 4 BEGIN
206 0205 4 RAB[RAB$B_RAC] = RAB$C_SEQ;
207 0206 4 STATUS = $GET (RAB = RAB);
208 0207 4 IF .STATUS EQLU RMSS_EOF OR .STATUS EQLU RMSS_OK_LIM
209 0208 4 THEN
210 0209 4 STATUS = SSS_NOSUCHID;
211 0210 4 IF NOT .STATUS
212 0211 4 THEN
213 0212 5 BEGIN
214 0213 5 EXESS$FINISH_RDB (.LOC_CONTXT);
215 0214 5 LEAVE RDB_OPEN;
216 0215 5 END
217 0216 3 END;
218 0217 3
219 0218 3 IF .LOC_ID NEQU 0 THEN .LOC_ID = .REC_BUFFER[KGB$L_IDENTIFIER];
220 0219 3 IF .LOC_ATTRIB NEQU 0 THEN .LOC_ATTRIB = .REC_BUFFER[KGB$L_ATTRIBUTES];
221 0220 3
222 0221 3 STATUS = SSS_NORMAL;
223 0222 3 END;
224 0223 2
225 0224 2 ! Close the rights database if there is no image
226 0225 2 !
227 0226 2
228 0227 2 IF .CLOSE THEN EXE$CLOSE_RDB();
229 0228 2 RETURN .STATUS
230 0229 1 END;

```

! End of routine SYSS\$FIND_HELD

```

.TITLE FINDHELD FINDHELD - SYSS$FIND_HELD loadable syst
em servic
.IDENT \V04-000\
.EXTRN EXE$OPEN_RDB, EXE$CLOSE_RDB
.EXTRN EXESS$FINISH_RDB
.EXTRN SYSS$GET
.PSECT $CODE$,NOWRT,2
.ENTRY SYSS$FIND_HELD, Save R2,R3,R4,R5,R6,R7,R8,- ; 0083
R9,R10,RT1
MOVAB -96(SP), SP ; 0143
MOVL HOLDER, LOC HOLDER ; 0144
PROBER #0, #8, (LOC HOLDER)
BEQL 5$ ;
MOVQ (LOC HOLDER), HOLDER_ID ; 0145
CMLPL HOLDER_ID, #1073741823 ; 0147
BGTRU 1$ ;
TSTL HOLDER_ID+4 ;
BEQL 2$ ;
MOVZWL #8740, R0 ; 0149

```

				OFFC 00000
	SE	A0	AE	9E 00002
	50	04	AC	D0 00006
60	08		00	0C 0000A
			4F	13 0000E
	58		60	7D 00010
	3FFFFFFF	8F	58	AE D1 00014
			05	1A 0001C
		5C	AE	D5 0001E
			06	13 00021
	50	2224	8F	3C 00023 1\$:

FINDHELD
V04-000

FINDHELD - SYSS\$FIND_HELD loadable system servic
H 6
16-Sep-1984 01:58:06
14-Sep-1984 12:40:51

VAX-11 Bliss-32 V4.0-742
[LOADSS.SRC]FINDHELD.B32;1

Page 7
(2)

00000000G	00		01	FB	000DB	CALLS	#1, SYSS\$GET	:	
	52		50	D0	000E2	MOVL	R0, STATUS	:	
0001827A	8F		52	D1	000E5	CMPL	STATUS, #98938	:	0207
			09	13	000EC	BEQL	11\$:	
00018051	8F		52	D1	000EE	CMPL	STATUS, #98385	:	
			05	12	000F5	BNEQ	12\$:	
	52	21EC	8F	3C	000F7	MOVZWL	#8684, STATUS	:	0209
	0B		52	E8	000FC	BLBS	STATUS, 13\$:	0210
			56	DD	00CFF	PUSHL	LOC_CONXT	:	0213
00000000G	9F		01	FB	00101	CALLS	#1, @#EXE\$\$FINISH_RDB	:	
			11	11	00108	BRB	16\$:	0214
	04		5B	E9	0010A	BLBC	R11, 14\$:	0218
	69	04	AE	D0	0010D	MOVL	REC_BUFFER, (LOC_ID)	:	
	04		5A	E9	00111	BLBC	R10, 15\$:	0219
	68	08	AE	D0	00114	MOVL	REC_BUFFER+4, (LOC_ATTRIB)	:	
	52		01	D0	00118	MOVL	#1, STATUS	:	0221
	07		6E	E9	0011B	BLBC	CLOSE, 17\$:	0227
00000000G	9F		00	FB	0011E	CALLS	#0, @#EXE\$CLOSE_RDB	:	
	50		52	D0	00125	MOVL	STATUS, R0	:	0228
			04	00128	RET			:	0229

; Routine Size: 297 bytes, Routine Base: \$CODE\$ + 0000

FINDHELD
V04-000

FINDHELD - SYSS\$FIND_HELD loadable system servic 1 6
16-Sep-1984 01:58:06
14-Sep-1984 12:40:51

VAX-11 Bliss-32 V4.0-742
[LOADSS.SRC]FINDHELD.B32;1

Page 8
(3)

: 232
: 233
0230 1 END
0231 0 ELUDOM

PSECT SUMMARY

Name Bytes Attributes
\$CODE\$ 297 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	51 0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:FINDHELD/OBJ=OBJ\$:FINDHELD MSRCS\$:FINDHELD/UPDATE=(ENHS\$:FINDHELD)

: Size: 297 code + 0 data bytes
: Run Time: 00:09.7
: Elapsed Time: 00:23.6
: Lines/CPU Min: 1434
: Lexemes/CPU-Min: 18211
: Memory Used: 133 pages
: Compilation Complete

