


```

CCCCCCCC 000000 BBBB8888 FFFFFFFF IIIIII NN NN DDDDDDD NN NN AAAAAA
CCCCCCCC 000000 BBBB8888 FFFFFFFF IIIIII NN NN DDDDDDD NN NN AAAAAA
CC        00      00  BB      BB  FF      II      NN      NN  DD      DD  NN      NN  AA      AA
CC        00      00  BB      BB  FF      II      NN      NN  DD      DD  NN      NN  AA      AA
CC        00      00  BB      BB  FF      II      NNNN     NN  DD      DD  NNNN     NN  AA      AA
CC        00      00  BB      BB  FF      II      NNNN     NN  DD      DD  NNNN     NN  AA      AA
CC        00      00  BBBB8888 FFFFFFFF IIIIII NN NN NN  DD      DD  NN NN NN  AA      AA
CC        00      00  BBBB8888 FFFFFFFF IIIIII NN NN NN  DD      DD  NN NN NN  AA      AA
CC        00      00  BB      BB  FF      II      NN NN NN  DD      DD  NN NN NN  AA      AA
CC        00      00  BB      BB  FF      II      NN NN NN  DD      DD  NN NN NN  AA      AA
CC        00      00  BB      BB  FF      II      NN NN NN  DD      DD  NN NN NN  AA      AA
CC        00      00  BB      BB  FF      II      NN NN NN  DD      DD  NN NN NN  AA      AA
CC        00      00  BBBB8888 FFFFFFFF IIIIII NN NN NN  DD      DD  NN NN NN  AA      AA
CCCCCCCC 000000 BBBB8888 FFFFFFFF IIIIII NN NN DDDDDDD NN NN AAAAAA
CCCCCCCC 000000 BBBB8888 FFFFFFFF IIIIII NN NN DDDDDDD NN NN AAAAAA

```

```

LL        IIIIII SSSSSSSS
LL        IIIIII SSSSSSSS
LL        II      SS
LL        II      SS
LL        II      SS
LL        II      SS
LL        II      SSSSSS
LL        II      SSSSSS
LL        II      SS
LL        II      SS
LL        II      SS
LL        II      SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```

```

1 0001 0 MODULE COBS$FIND_NAME (
2 0002 0   IDENT = '1-013'           ! file: COBFINDNA.B32 EDIT:LB1013
3 0003 0   ) =
4 0004 1 BEGIN
5 0005 1
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:      COBOL SUPPORT
33 0033 1
34 0034 1 ABSTRACT:      This procedure will search lists of program names for
35 0035 1 a routine name match to the name found in the input
36 0036 1 parameter "DESC".
37 0037 1
38 0038 1
39 0039 1
40 0040 1 ENVIRONMENT:   Vax-11 User Mode
41 0041 1 NOTE this module cannot be in sharable library.
42 0042 1
43 0043 1 AUTHOR: M.L.Jack , CREATION DATE: 29-JUL-1979
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1 1-001 - Original. MLJ 29-JUL-1979
48 0048 1 1-002 - Added boilerplate and comments. RKR 18-JULY-1979
49 0049 1 1-003 - Make LIBRARY and REQUIRE file compatible with rest of system.
50 0050 1 RKR 19-SEPT-79
51 0051 1 1-004 - Rewrite name string analysis to catch leading space case.
52 0052 1 Current logic will return status of zero if any of the
53 0053 1 following occur:
54 0054 1 . Leading spaces or tabs are present in the string
55 0055 1 . Resulting string with trailing spaces or tabs
56 0056 1 exceed 30 chars.
57 0057 1 . Resulting string cannot be found in table

```

```

: 58      0058  1  : Imbedded spaces or tabs stop the scan the same way trailing
: 59      0059  1  : spaces and tabs do, and will probably yield look-up failure
: 60      0060  1  : RKR 2-OCT-79
: 61      0061  1  : 1-005 - Revise algorithm to simply throw away leading spaces and tabs
: 62      0062  1  : and continue scanning. RKR 3-OCT-79
: 63      0063  1  : 1-006 - Also allow %C'0' TO %C'9' as legal chars in name. (Oversight)
: 64      0064  1  : RKR 04-OCT-79
: 65      0065  1  : 1-007 - Accomodate new structure for module name data table.
: 66      0066  1  : RKR 19-OCT-79
: 67      0067  1  : 1-008 - Change name extraction logic to behave like GENEXTERN.B32
: 68      0068  1  : RKR 19-OCT-79
: 69      0069  1  : 1-009 - Add error for syntactically invalid name. RKR 21-OCT-79
: 70      0070  1  : 1-010 - Remove all signalling. RKR 29-OCT-79
: 71      0071  1  : 1-011 - Remove definition of OBJ$C_SYMSIZ. SBL 5-Dec-1979
: 72      0072  1  : 1-012 - Introduce a new symbol LOCAL_SYMSIZ to replace role of
: 73      0073  1  : former symbol OBJ$C_SYMSIZ. At sometime when the libraries
: 74      0074  1  : are universally straightened out we can switch back to
: 75      0075  1  : OBJ$C_SYMSIZ and allow it to be defined via library.
: 76      0076  1  : (By the way, its value should be 31)
: 77      0077  1  : RKR 06-DEC-79
: 78      0078  1  : 1-013 - Added abstract, functional description, comments and made
: 79      0079  1  : cosmetic changes. Changed hardcoded codes for tabs and
: 80      0080  1  : blanks into literals to enhance readability. Also added
: 81      0081  1  : code that checks to make sure that if a second input
: 82      0082  1  : parameter exists, that it is non-zero. Changed code that
: 83      0083  1  : converted lowercase characters to uppercase characters.
: 84      0084  1  : LB 3-MAR-81
: 85      0085  1  :
: 86      0086  1  : --
: 87      0087  1  :

```

```

89      0088 1  !+
90      0089 1  ! SWITCHES
91      0090 1  !-
92      0091 1
93      0092 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
94      0093 1
95      0094 1  !+
96      0095 1  ! LINKAGES
97      0096 1  !     NONE
98      0097 1  !-
99      0098 1
100     0099 1  !+
101     0100 1  ! TABLE OF CONTENTS:
102     0101 1  !-
103     0102 1
104     0103 1 FORWARD ROUTINE
105     0104 1
106     0105 1     COB$$FIND_NAME ;
107     0106 1
108     0107 1  !+
109     0108 1  ! INCLUDE FILES
110     0109 1  !-
111     0110 1
112     0111 1 LIBRARY 'RTLSTARLE';
113     0112 1 REQUIRE 'RTLIN:COBDEF';
114     0554 1
115     0555 1  !+
116     0556 1  ! MACROS
117     0557 1  !     NONE
118     0558 1  !-
119     0559 1
120     0560 1  !+
121     0561 1  ! EQUATED SYMBOLS
122     0562 1  !-
123     0563 1
124     0564 1 LITERAL
125     0565 1     TRUE = 1,
126     0566 1     FALSE = 0,
127     0567 1     TAB = %0'11',
128     0568 1     BLANK = %0'40',
129     0569 1     LOCAL_SYMSIZ = 31 ;
130     0570 1
131     0571 1
132     0572 1
133     0573 1
134     0574 1
135     0575 1
136     0576 1  !+
137     0577 1  ! PSECT DECLARATIONS:
138     0578 1  !-
139     0579 1
140     0580 1 PSECT
141     0581 1     GLOBAL= COB$NAMES_____1(PIC,SHARE,NOWRITE,ADDRESSING_MODE(LONG_RELATIVE));
142     0582 1 GLOBAL
143     0583 1     COB$GZ_NAMES_1: VECTOR[0];
144     0584 1 PSECT
145     0585 1     GLOBAL= COB$NAMES_____3(PIC,SHARE,NOWRITE,ADDRESSING_MODE(LONG_RELATIVE));

```

! Contains all RTL routines
! COBOL specific RTL macros and literals

! TAB character
! BLANK character
! When libraries get consistent on
! various development machines, this
! definition should be removed and
! the corresponding references in the
! code should be to the library-defined
! symbol OBJ\$C_SYMSIZ

COB\$IND_NAME
1-013

I 16
16-Sep-1984 00:08:22
14-Sep-1984 12:10:46

VAX-11 Bliss-32 V4.0-742
[COBRTL.SRC]COB\$INDNA.B32;1

Page 4
(2)

```
: 146      0586 1 GLOBAL  
: 147      0587 1  
: 148      0588 1 COB$GZ_NAMES_2: VECTOR[0];  
: 149      0589 1  
: 150      0590 1 PSECT  
: 151      0591 1 CODE= _COB$CODE(PIC,SHARE),  
: 152      0592 1 PLIT=  _COB$CODE;
```

```

154 0593 1 GLOBAL ROUTINE COB$FIND_NAME (DESC,LLIST)=
155 0594 1
156 0595 1 +-
157 0596 1 FUNCTIONAL DESCRIPTION:
158 0597 1
159 0598 1 This routine will search through the list of local program names
160 0599 1 to find a match to the name that is passed in the "DESC"
161 0600 1 input parameter. If no match is found, it searches through
162 0601 1 the list of global program names (designated by the COB$GZ_NAMES_2
163 0602 1 structure). If a match is found, then it returns a pointer to
164 0603 1 the COB$NAMES entry; otherwise, it returns a value of zero.
165 0604 1
166 0605 1 CALLING SEQUENCE:
167 0606 1
168 0607 1 COB$FIND_NAME (desc.rt.ds, llist.rr.r)
169 0608 1
170 0609 1 FORMAL PARAMETERS:
171 0610 1
172 0611 1 DESC.rt.ds Address of descriptor of name string
173 0612 1
174 0613 1 LLIST.rr.r Pointer to counted list of accessible names
175 0614 1
176 0615 1 IMPLICIT INPUTS:
177 0616 1
178 0617 1 NONE
179 0618 1
180 0619 1 IMPLICIT OUTPUTS:
181 0620 1
182 0621 1 NONE
183 0622 1
184 0623 1 ROUTINE VALUE:
185 0624 1
186 0625 1 Pointer to COB$NAMES entry, or 0 if not found
187 0626 1
188 0627 1 SIDE EFFECTS:
189 0628 1
190 0629 1 NONE
191 0630 1
192 0631 1 --
193 0632 1
194 0633 2 BEGIN
195 0634 2
196 0635 2 MAP
197 0636 2 DESC: REF BLOCK[,BYTE], ! Pointer to descriptor for name string
198 0637 2 LLIST: REF VECTOR; ! Pointer to counted list of local programs
199 0638 2
200 0639 2 BUILTIN
201 0640 2 ACTUALCOUNT, !\Returns # of actual parameters
202 0641 2 !/passed to this routine
203 0642 2 ACTUALPARAMETER; !\Returns the value of the i'th
204 0643 2 !/actual parameter that was passed
205 0644 2
206 0645 2 LOCAL
207 0646 2 BUFFER: VECTOR[LOCAL_SYMSIZ,BYTE], ! Buffer for input name
208 0647 2 NAMELENGTH, ! Length of input name
209 0648 2 NAMEPOINTER: REF VECTOR[,BYTE], ! Pointer to input name
210 0649 2 A, ! Offset of first nonblank

```

```

211      0650      2      P: REF BLOCK [,BYTE],           ! Pointer for searching
212      0651      2      L,           ! Length of finished literal
213      0652      2      Q: REF VECTOR[,BYTE];       ! Pointer to ASCII name string
214      0653      2
215      0654      2      NAMELENGTH = .DESC[DSCSW_LENGTH]; ! Length of input name
216      0655      2      NAMEPOINTER = .DESC[DSCSA_POINTER]; ! Pointer to input name
217      0656      2
218      0657      2
219      0658      2
220      0659      2
221      0660      2      L = 0;           ! Init length of finished literal
222      0661      2
223      0662      2
224      0663      2      !+
225      0664      2      ! Strip trailing blanks and tabs.
226      0665      2      !-
227      0666      2
228      0667      2      DECR I FROM .NAMELENGTH-1 TO 0 DO
229      0668      2      BEGIN
230      0669      2      IF .NAMEPOINTER[I] NEQ BLANK AND .NAMEPOINTER[I] NEQ TAB
231      0670      2      THEN
232      0671      2      BEGIN
233      0672      2      L = .I + 1;           ! \Length of name without
234      0673      2      ! /trailing blanks and spaces
235      0674      2      EXITLOOP;           ! Found a valid char - exit the loop
236      0675      2      END;
237      0676      2      END;
238      0677      2
239      0678      2
240      0679      2      IF .L EQL 0
241      0680      2      THEN
242      0681      2      RETURN 0           ! Invalid parameter - no input name
243      0682      2      ELSE
244      0683      2      BEGIN
245      0684      2
246      0685      2      !+
247      0686      2      ! Strip leading spaces and tabs.
248      0687      2      !-
249      0688      2
250      0689      2      A = 0;
251      0690      2      WHILE .NAMEPOINTER[A] EQL BLANK OR .NAMEPOINTER[A] EQL TAB DO
252      0691      2      BEGIN
253      0692      2      L = .L - 1;           ! \Decr name length for every
254      0693      2      ! /occurrence of blanks or tabs
255      0694      2      A = .A + 1;           ! Advance the index
256      0695      2      END;
257      0696      2
258      0697      2
259      0698      2      !+
260      0699      2      ! Ensure that the result, with leading and trailing spaces
261      0700      2      ! removed, is not too long.
262      0701      2      !-
263      0702      2
264      0703      2      IF .L GTR LOCAL_SYMSIZ
265      0704      2      THEN
266      0705      2      RETURN 0 ;           ! Invalid symbol name
267      0706      2

```



```

268 0707
269 0708
270 0709
271 0710
272 0711
273 0712
274 0713
275 0714
276 0715
277 0716
278 0717
279 0718
280 0719
281 0720
282 0721
283 0722
284 0723
285 0724
286 0725
287 0726
288 0727
289 0728
290 0729
291 0730
292 0731
293 0732
294 0733
295 0734
296 0735
297 0736
298 0737
299 0738
300 0739
301 0740
302 0741
303 0742
304 0743
305 0744
306 0745
307 0746
308 0747
309 0748
310 0749
311 0750
312 0751
313 0752
314 0753
315 0754
316 0755
317 0756
318 0757
319 0758
320 0759
321 0760
322 0761
323 0762
324 0763

```

```

+
Now convert and check the characters for legality in the
object language. The valid characters for COBOL string names
include all uppercase characters, 0-9, dollar sign ($) and
underscore (_). Note that all lowercase characters are converted
to uppercase and that hyphens are converted to the underscore
character.
-

INCR I FROM 0 TO .L-1 DO
  BEGIN
    LOCAL
      C;

      C = .NAMEPOINTER[.I+.A];
      SELECTONE .C OF
        SET
          [XC'-' ]:           ! Convert hyphen to underscore
            C = XC'_' ;
          [XC'a' TO XC'z' ]:  ! Convert lowercase to uppercase
            C = .C - BLANK;   ! Just subtract %0'40' to get uppercase
          [XC'A' TO XC'Z' ,
           XC'O' TO XC'9' ,
           XC'$' , XC'_' ]:
            0;               ! Characters are valid - do nothing
          [OTHERWISE]:      ! Error - invalid characters
            RETURN 0 ;
        TES;

      ++
      Complete name is copied into 'BUFFER'
      one character at a time.
      --

      BUFFER[.I] = .C
      END;
    END;

+
If resultant string exists, try looking it up.

If there is more than 1 input parameter and the second parameter
is non-zero, then search through the list of local programs.
Otherwise, try the list of global programs. Note that when
searching through the list of local programs and a match is not
found, the default is to search through the list of global
programs.

IF (ACTUALCOUNT() GTR 1) AND (ACTUALPARAMETER (ACTUALCOUNT()) NEQ 0)
  THEN
    INCR I FROM 1 TO .LLIST[0] DO
      BEGIN
        P = .LLIST[.I];           ! Fetch addr of name entry block
        Q = .P[COBS$A_NAM_NAME]; ! Fetch 1st byte of counted name string

```

```

325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360

```

```

0764 3 IF .Q[0] EQL .L ! If length matches
0765 3 THEN
0766 4 BEGIN
0767 4 IF CH$EQL (.L,Q[1], .L, BUFFER) ! If string matches
0768 4 THEN
0769 4 RETURN .P ! Return addr of this name entry block
0770 4 ELSE
0771 4 0 ! No match occurred
0772 3 END;
0773 2 END;
0774 2
0775 2
0776 2
0777 2
0778 2
0779 2
0780 2
0781 2
0782 2
0783 2
0784 3
0785 3
0786 3
0787 3
0788 4
0789 4
0790 4
0791 4
0792 4
0793 4
0794 3
0795 3
0796 2
0797 2
0798 2
0799 1

```

Now look for global programs. Note that
COB\$GZ_NAMES_1 points to the beginning of the
COB\$GZ_NAMES_2 structure while COB\$GZ_NAMES_2
points to the end of the structure.

```

P = COB$GZ_NAMES_1 ; ! 1st (if any) entry
WHILE .P LSSA COB$GZ_NAMES_2 DO
BEGIN
Q = .P[COB$A_NAM_NAME] ; ! Ptr to 1st byte of counted name string
IF .Q[0] EQL .L ! If length matches
THEN
BEGIN
IF CH$EQL (.L, Q[1], .L, BUFFER ) ! If string matches
THEN
RETURN .P ! Retn addr of this name entry block
ELSE
0 ! No match found
END ;
P = .P + .P[COB$L_NAM_LEN] ; ! Advance to next entry
END ;
RETURN 0;
END ; ! End of routine COB$$FIND_NAME

```

```

.TITLE COB$FIND_NAME
.IDENT \1-013\
.PSECT COB$NAMES_____3,NOWRT,NOEXE, SHR, PIC,2
0000 COB$GZ_NAMES_2::
.BLK 0
.PSECT COB$NAMES_____1,NOWRT,NOEXE, SHR, PIC,2
0000 COB$GZ_NAMES_1::
.BLK 0
.PSECT _COB$CODE,NOWRT, SHR, PIC,2
SE 00FC 0000 .ENTRY COB$$FIND_NAME, Save R2,R3,R4,R5,R6,R7 : 0593
20 C2 00002 SUBL2 #32, SP ;

```

	51	04	AC	D0	00005		MOVL	DESC, R1	0654
	50		61	3C	00009		MOVZWL	(R1), NAMELENGTH	
	52	04	A1	D0	0000C		MOVL	4(R1), NAMEPOINTER	0655
			54	D4	00010		CLRL	L	0660
			12	11	00012		BRB	2\$	0669
	20		6042	91	00014	1\$:	CMPB	(I)[NAMEPOINTER], #32	
			0C	13	00018		BEQL	2\$	
	09		6042	91	0001A		CMPB	(I)[NAMEPOINTER], #9	
			06	13	0001E		BEQL	2\$	
	54	01	A0	9E	00020		MOVAB	1(R0), L	0672
			03	11	00024		BRB	3\$	0671
	EB		50	F4	00026	2\$:	SOBGEQ	I, 1\$	0667
			54	D5	00029	3\$:	TSTL	L	0679
			03	12	0002B		BNEQ	5\$	
			00D5	31	0002D	4\$:	BRW	22\$	
			53	D4	00030	5\$:	CLRL	A	0689
	20		6342	91	00032	6\$:	CMPB	(A)[NAMEPOINTER], #32	0690
			06	13	00036		BEQL	7\$	
	09		6342	91	00038		CMPB	(A)[NAMEPOINTER], #9	
			06	12	0003C		BNEQ	8\$	
			54	D7	0003E	7\$:	DECL	L	0692
			53	D6	00040		INCL	A	0694
			EE	11	00042		BRB	6\$	0690
	1F		54	D1	00044	8\$:	CMPL	L #31	0703
			E4	14	00047		BGTR	4\$	
	50		01	CE	00049		MNEGL	#1, I	0723
			58	11	0004C		BRB	15\$	
51	50		53	C1	0004E	9\$:	ADDL3	A, I, R1	
	51		6142	9A	00052		MOVZBL	(R1)[NAMEPOINTER], C	
	2D		51	D1	00056		CMPL	C, #45	0726
			06	12	00059		BNEQ	10\$	
	51	SF	8F	9A	0005B		MOVZBL	#95, C	0727
			41	11	0005F		BRB	14\$	
00000061	8F		51	D1	00061	10\$:	CMPL	C, #97	0728
			0E	19	00068		BLSS	11\$	
0000007A	8F		51	D1	0006A		CMPL	C, #122	
			05	14	00071		BGTR	11\$	
	51		20	C2	00073		SUBL2	#32, C	0729
			2A	11	00076		BRB	14\$	
	24		51	D1	00078	11\$:	CMPL	C, #36	0730
			25	13	0007B		BEQL	14\$	
	30		51	D1	0007D		CMPL	C, #48	
			05	19	00080		BLSS	12\$	
	39		51	D1	00082		CMPL	C, #57	
			1B	15	00085		BLEQ	14\$	
00000041	8F		51	D1	00087	12\$:	CMPL	C, #65	
			09	19	0008E		BLSS	13\$	
0000005A	8F		51	D1	00090		CMPL	C, #90	
			09	15	00097		BLEQ	14\$	
0000005F	8F		51	D1	00099	13\$:	CMPL	C, #95	
			63	12	000A0		BNEQ	22\$	
	6E40		51	90	000A2	14\$:	MOVB	C, BUFFER[1]	0743
A4	50		54	F2	000A6	15\$:	AOBLSS	L, I, 9\$	
	01		6C	91	000AA		CMPB	(AP), #1	0758
			28	1B	000AD		BLEQU	18\$	
	50		6C	9A	000AF		MOVZBL	(AP), R0	
			6C40	D5	000B2		TSTL	(AP)[R0]	

				20	13	000B5		BEQL	18\$		
				56	D4	000B7		CLRL	I		0764
				17	11	000B9		BRB	17\$		
		55	08	BC46	D0	000BB	16\$:	MOVL	@LLISTE[I], P		0762
54		57	04	A5	D0	000C0		MOVL	4(P), Q		0763
	67	08		00	ED	000C4		CMPZV	#0, #8, (Q), L		0764
				07	12	00...9		BNEQ	17\$		
	6E	01	A7	54	29	000CB		CMPC3	L, 1(Q), BUFFER		0767
				2A	13	000D0		BEQL	20\$		
	E4			56	08	BC	F3 000D2 17\$:	AOBLEQ	@LLIST, I, 16\$		0760
		55	00000000'	EF	9E	000D7	18\$:	MOVAB	COB\$GZ_NAMES_1, P		0782
		50	00000000'	EF	9E	000DE	19\$:	MOVAB	COB\$GZ_NAMES_2, RO		0783
		50		55	D1	000E5		CP	P, RO		
				1B	1E	000E8		BGEQU	22\$		
		57	04	A5	D0	000EA		MOVL	4(P), Q		0785
54	67	08		00	ED	000EE		CMPZV	#0, #8, (Q), L		0786
				0B	12	000F3		BNEQ	21\$		
	6E	01	A7	54	29	000F5		CMPC3	L, 1(Q), BUFFER		0789
				04	12	000FA		BNEQ	21\$		
		50		55	D0	000FC	20\$:	MOVL	P, RO		0791
					04	000FF		RET			
		55		65	C0	00100	21\$:	ADDL2	(P), P		0795
				D9	11	00103		BRB	19\$		0783
				50	D4	00105	22\$:	CLRL	RO		0799
					04	00107		RET			

: Routine Size: 264 bytes, Routine Base: _COB\$CODE + 0000

: 361 0800 1 END
: 362 0801 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
COB\$NAMES-----1	0	NOVEC,NOWRT, RD ,NOEXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
COB\$NAMES-----3	0	NOVEC,NOWRT, RD ,NOEXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
_COB\$CODE	264	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]STARLET.L32;1	9776	2	0	581	00:00.8

COB\$FIND_NAME
1-013

D 1
16-Sep-1984 00:08:22
14-Sep-1984 12:10:46

VAX-11 BLISS-32 V4.0-742
[COBRTL.SRC]COB\$FINDNA.B32;1

Page 11
(3)

COMMAND QUALIFIERS

```
:  
:      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:COB$FINDNA/OBJ=OBJS:COB$FINDNA MSRCS:COB$FINDNA/UPDATE=(ENHS:COB$FINDNA  
:      )  
:  
: Size:          264 code + 0 data bytes  
: Run Time:      00:05.9  
: Elapsed Time: 00:20.9  
: Lines/CPU Min: 8145  
: Lexemes/CPU-Min: 22942  
: Memory Used:  119 pages  
: Compilation Complete
```

0062 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

COBDIVQ LIS

COBDEBIT LIS

COBDISPLA LIS

COBESGEN LIS

COBERROR LIS

COBHANDL LIS

COBEXPI LIS

COBFINDNA LIS

COBBEXCE LIS

COBEXP1 LIS

