

[illegible]

3-

Sy

LI

LI
LILI
LI
LILI
LI

LI

LI
LILI
LILI
LI

LI

LI
LI

LI

LI
LI

LI

LI
LI

LI

11

LI

LI
LI

LI

LI
LI

21

LI

LI
LI

22

11

LI
LI

LI

LI
LI

LI

```

LL      IIIIII  BBBB BBBB LL      000000  000000  KK      KK  UU      UU  P P P P P P P P
LL      IIIIII  BBBB BBBB LL      000000  000000  KK      KK  UU      UU  P P P P P P P P
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LL      II      BB      BB LL      00      00  00      00  KK      KK  UU      UU  PP      PP
LLLLLLLLLLLL IIIIII  BBBB BBBB LLLLLLLLLLLL 000000  000000  KK      KK  UUUUUUUUUU PP
LLLLLLLLLLLL IIIIII  BBBB BBBB LLLLLLLLLLLL 000000  000000  KK      KK  UUUUUUUUUU PP
                                     ....
                                     ....
                                     ....
                                     ....

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
LL      II      SSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS

```



```
1 0001 0 MODULE LIB$LOOKUP_KEY ( ! Keyword lookup routine
2 0002 0
3 0003 0 IDENT = '1-012' ! File: LIBLOOKUP.B32 Edit: RKR1012
4 0004 0
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
19 0019 1 * TRANSFERRED. *
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
23 0023 1 * CORPORATION. *
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 ++
33 0033 1 FACILITY: General Utility Procedure Library
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine attempts to match a caller-specified character
38 0038 1 string with a table of keywords built by the caller.
39 0039 1
40 0040 1 ENVIRONMENT: User mode, AST re-entrant. Non-shared library
41 0041 1
42 0042 1 AUTHOR: Ward Clark, CREATION DATE: 10 January 1978
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1 Jonathan M. Taylor - 5-Mar-78
46 0046 1
47 0047 1 0-2 - Fixed call to LIB$SCOPY_R_DX after changing parameter
48 0048 1 order. JMT 5-Mar-78
49 0049 1 0-03 - Change to STARLET library. DGP 20-Apr-78
50 0050 1 0-04 - Change REQUIRE files for VAX system build. DGP 28-Apr-78
51 0051 1 0-05 - Change STARLET to RTLSTARLE to avoid conflicts. DGP 1-May-78
52 0052 1 0-06 - Addressing mode general for LIB$SCOPY. TNH 17-June-78
53 0053 1 0-08 - Change file name to LIBLOOKUP.B32. JBS 14-NOV-78
54 0054 1 1-001 - Update version number. JBS 16-NOV-78
55 0055 1 1-002 - Remove REQUIRE of LIBMAC. Only LIBLOOKUP is using it so
56 0056 1 put its text directly in this file. JBS 11-DEC-78
57 0057 1 1-003 - Add REQUIRE of LIBMSG to define error symbols. JBS 11-DEC-78
```

```

: 58      0058 1 1-004 - Change LIB$$ to STR$. JBS 23-MAY-1979
: 59      0059 1 1-005 - Change call to STR$COPY. JBS 16-JUL-1979
: 60      0060 1 1-006 - Declare message symbols as externals, add OUTLEN, and do some
: 61      0061 1      minor cleanups to the text. JBS 18-SEP-1979
: 62      0062 1 1-007 - Remove $LIB_KEY_TABLE, moved to RTLMACB32.REQ.
: 63      0063 1      JBS 19-DEC-1979
: 64      0064 1 1-008 - Use handler to translate signals. RW 22-Jan-1980
: 65      0065 1 1-009 - Fix bug where STR$COPY_R was being called with length passed
: 66      0066 1      by value rather than by reference. SBL 11-Mar-1980
: 67      0067 1 1-010 - Enhance to recognize additional classes of string descriptors
: 68      0068 1      by invoking LIB$ANALYZE_SDESC_R3 to extract length and
: 69      0069 1      address of 1st data byte from descriptor.
: 70      0070 1      Change call to STR$COPY_R to LIB$SCOPY_R_DX.
: 71      0071 1      This eliminates the need to establish a handler and the need
: 72      0072 1      to convert STR$ statuses to LIB$ statuses.
: 73      0073 1      RKR 28-MAY-1981.
: 74      0074 1 1-011 - Add special-case code to process string descriptors that
: 75      0075 1      'read' like fixed string descriptors. RKR 7-OCT-1981.
: 76      0076 1 1-012 - Redirect jsb's from LIB$ANALYZE_SDESC_R3 to
: 77      0077 1      LIB$ANALYZE_SDESC_R2. Use LIB$SCOPY_R_DX6 to do copying.
: 78      0078 1      RKR 18-NOV-1981.
: 79      0079 1      --
: 80      0080 1 <BLF/PAGE>
```



```
.. 82      0081 1  |
.. 83      0082 1  | SWITCHES:
.. 84      0083 1  |
.. 85      0084 1  |
.. 86      0085 1  | SWITCHES ADDRESSING MODE
.. 87      0086 1  |             (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
.. 88      0087 1  |
.. 89      0088 1  |
.. 90      0089 1  | LINKAGES
.. 91      0090 1  |
.. 92      0091 1  | REQUIRE 'RTLIN:STRLNK';           ! Linkage for LIB$ANALYZE_SDESC_R2
.. 93      0276 1  |
.. 94      0277 1  |
.. 95      0278 1  | TABLE OF CONTENTS:
.. 96      0279 1  |
.. 97      0280 1  |
.. 98      0281 1  | FORWARD ROUTINE
.. 99      0282 1  |     LIB$LOOKUP_KEY;           ! Keyword table scanning routine
100      0283 1  |
101      0284 1  |
102      0285 1  | INCLUDE FILES:
103      0286 1  |
104      0287 1  |
105      0288 1  | LIBRARY 'RTLSTARLE';           ! VAX/VMS Literals
106      0289 1  |
107      0290 1  | REQUIRE 'RTLIN:RTLPSECT';       ! Define DECLARE_PSECTS macro
108      0385 1  |
109      0386 1  |
110      0387 1  | MACROS:
111      0388 1  |
112      0389 1  | +
113      0390 1  | Macros used to access various elements of a keyword table built
114      0391 1  | by a $LIB_KEY_TABLE macro for use with the LIB$KEY_TABLE keyword
115      0392 1  | lookup routine.
116      0393 1  | -
117      0394 1  |
118      0395 1  | MACRO
119      M 0396 1  |     $LIB_KEY_LENGTH (TABLE_POINTER) =
120      0397 1  |     .(.TABLE_POINTER)<0,8> %;    ! Length of a keyword table string
121      M 0398 1  |     $LIB_KEY_STRING (TABLE_POINTER) =
122      0399 1  |     .TABLE_POINTER + 1 %;        ! Address of a keyword table string
123      M 0400 1  |     $LIB_KEY_VALUE (TABLE_POINTER) =
124      0401 1  |     .(TABLE_POINTER + 4) %;      ! Value corresponding to a keyword table entry
125      0402 1  |
126      0403 1  |
127      0404 1  | EQUATED SYMBOLS:
128      0405 1  |
129      0406 1  |     None
130      0407 1  |
131      0408 1  | PSECT DECLARATIONS:
132      0409 1  |
133      0410 1  | DECLARE_PSECTS (LIB);           ! This module belongs to the LIB facility
134      0411 1  |
135      0412 1  | OWN STORAGE:
136      0413 1  |
137      0414 1  |     None
138      0415 1  |
```

```
: 139      0416 1 ! EXTERNAL REFERENCES:
: 140      0417 1 !
: 141      0418 1 !
: 142      0419 1 EXTERNAL ROUTINE
: 143      0420 1   LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK, ! Extract length
: 144      0421 1                                     ! and address of
: 145      0422 1                                     ! 1st data byte
: 146      0423 1                                     ! from descriptor
: 147      0424 1   LIB$COPY_R_DX6 : STRING_JSB ; ! Copy a by-reference string to a
: 148      0425 1                                     ! descriptor of any type.
: 149      0426 1
: 150      0427 1 !+
: 151      0428 1 ! The following are the conditions returned by this module
: 152      0429 1 !-
: 153      0430 1
: 154      0431 1 EXTERNAL LITERAL
: 155      0432 1   LIB$_STRTRU, ! String truncated
: 156      0433 1   LIB$_AMBKEY, ! Multiple keyword match found
: 157      0434 1   LIB$_UNRKEY, ! No keyword match found
: 158      0435 1   LIB$_INVARG; ! Invalid argument(s)
: 159      0436 1
```



```
161 0437 1 GLOBAL ROUTINE LIB$LOOKUP_KEY (      ! Keyword table scanning routine
162 0438 1
163 0439 1     STRNG_DESC_ADDR,      ! Search string
164 0440 1     KEY_TABLE_ADDR,    ! Keyword table
165 0441 1     KEY_VALUE_ADDR,    ! Keyword value deposit
166 0442 1     area              ! area
167 0443 1     FULL_DESC_ADDR,   ! Full keyword deposit
168 0444 1     area              ! area
169 0445 1     OUTLEN            ! Number of bytes stored
170 0446 1                          ! in FULL_DESC_ADDR
171 0447 1
172 0448 1                          ) =
173 0449 1
174 0450 1
```

++
FUNCTIONAL DESCRIPTION:

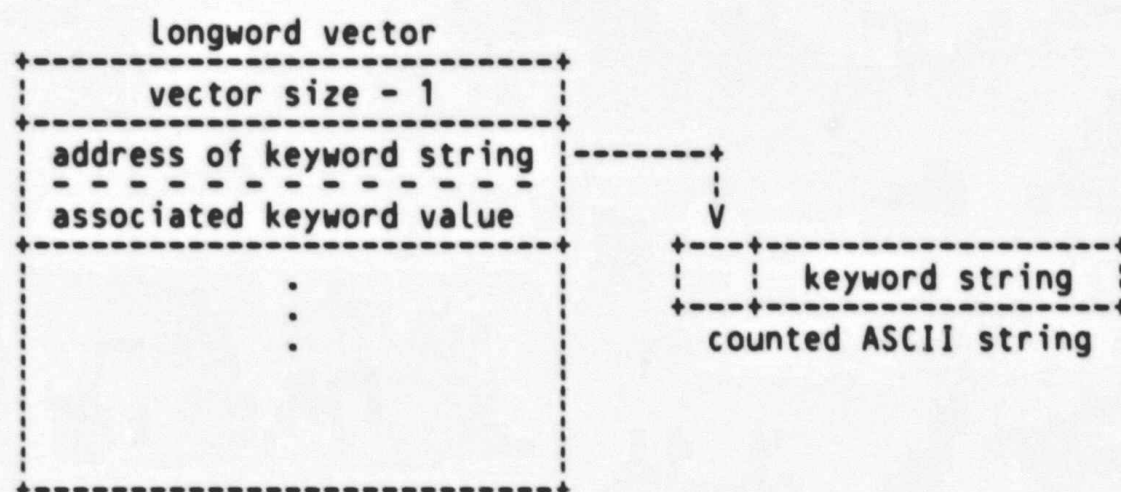
This keyword lookup routine scans a table of keywords (see below), attempting to find a keyword which matches a caller-specified keyword or keyword abbreviation.

When a keyword match is found, the following information is optionally returned to the caller:

- * longword value associated with the matched keyword (KEY_VALUE_ADDR)
- * full-keyword string (any descriptor type) (FULL_DESC_ADDR)

If an exact keyword match is found (i.e., the caller's search string is an unabbreviated keyword), no further processing is performed and a normal completion code is returned to the caller. Otherwise, after a match has been found, the rest of the keyword table is scanned. If an additional match is found, a "not enough characters" completion code is returned to the caller.

The keyword table, which the caller creates for this routine has the following structure:




```
218 0494 1 | where the "counted ASCII string" starts with a byte which is
219 0495 1 | the unsigned count of the number of ASCII characters which
220 0496 1 | follow.
221 0497 1 |
222 0498 1 | FORMAL PARAMETERS:
223 0499 1 |
224 0500 1 |     STRNG_DESC_ADDR.rt.dx - Address of search string descriptor
225 0501 1 |     KEY_TABLE_ADDR.rlu.ra - Address of keyword table
226 0502 1 |     [KEY_VALUE_ADDR.wlu.r] - Address of keyword value deposit area
227 0503 1 |                             (optional)
228 0504 1 |     [FULL_DESC_ADDR.wt.dx] - Address of full keyword deposit area
229 0505 1 |                             (optional)
230 0506 1 |     [OUTLEN.ww.r] - Number of bytes stored in FULL_DESC_ADDR
231 0507 1 |                     (optional)
232 0508 1 |
233 0509 1 | IMPLICIT INPUTS:
234 0510 1 |
235 0511 1 |     None
236 0512 1 |
237 0513 1 | IMPLICIT OUTPUTS:
238 0514 1 |
239 0515 1 |     None
240 0516 1 |
241 0517 1 | COMPLETION CODES:
242 0518 1 |
243 0519 1 |     $$$ NORMAL = Unique keyword match found
244 0520 1 |     LIB$_FATERRLIB = Fatal error in library
245 0521 1 |     LIB$_INVSTRDES = Invalid string descriptor
246 0522 1 |     LIB$_INVSIRMEM = Insufficient Virtual memory
247 0523 1 |     LIB$_STRTRU = String is truncated
248 0524 1 |     LIB$_AMBKEY = Multiple keyword match found (i.e., not enough
249 0525 1 |                  characters specified)
250 0526 1 |     LIB$_UNRKEY = No keyword match found
251 0527 1 |     LIB$_INVARG = Invalid arguments, not enough arguments, bad
252 0528 1 |                  keyword table or bad string descriptor
253 0529 1 |
254 0530 1 | SIDE EFFECTS:
255 0531 1 |
256 0532 1 |     None
257 0533 1 |
258 0534 1 | --
259 0535 1 |
260 0536 2 | BEGIN
261 0537 2 |
262 0538 2 | BUILTIN
263 0539 2 |     ACTUALCOUNT, ! Actual parameter count
264 0540 2 |     NULLPARAMETER; ! Test for presence of parameter
265 0541 2 |
266 0542 2 | MAP
267 0543 2 |     STRNG_DESC_ADDR : REF BLOCK [, BYTE], ! caller's search string
268 0544 2 |     FULL_DESC_ADDR : REF BLOCK [, BYTE], ! returned full key
269 0545 2 |     OUTLEN : REF VECTOR [1, WORD, UNSIGNED]; ! Number of bytes
270 0546 2 |                                           ! stored in
271 0547 2 |                                           ! FULL_DESC_ADDR
272 0548 2 |
273 0549 2 | BIND ! Redefine some routine arguments:
274 0550 2 |
```



```
275 0551 2 KEY_TABLE = .KEY_TABLE_ADDR : VECTOR, ! Caller's keyword table
276 0552 2 KEY_VALUE = .KEY_VALUE_ADDR; ! Caller's keyword
277 0553 2 ! value deposit area
278 0554 2
279 0555 2 LOCAL
280 0556 2 STRING_LEN, ! Length of search string
281 0557 2 STRING_ADDR, ! Address of search string
282 0558 2 MATCH, ! Index of a matched keyword table entry
283 0559 2 RETURN_CODE; ! Routine return code
284 0560 2
285 0561 2 !+
286 0562 2 !- Verify that the caller provided a proper argument list.
287 0563 2
288 0564 2
289 0565 2 IF (ACTUALCOUNT () LSSU 2) ! If less than 2 arguments were provided,
290 0566 2 THEN
291 0567 2 RETURN (LIB$_INVARG); ! return an error code to the caller.
292 0568 2
293 0569 2 !+
294 0570 2 !- Prepare to scan the caller's keyword table.
295 0571 2
296 0572 2 RETURN_CODE = LIB$_UNRKEY; ! Initially assume no keyword match
297 0573 2 ! exists.
298 0574 2 !+
299 0575 2 !- Extract length and address of 1st data byte of search string.
300 0576 2 If error results from attempt to extract, return that error.
301 0577 2
302 0578 2 IF .STRNG_DESC_ADDR [DSC$B_CLASS] GTRU DSC$K_CLASS_D
303 0579 2 THEN ! Use generalized extract
304 0580 2 BEGIN
305 0581 2 LOCAL RET_STATUS ;
306 0582 2 RET_STATUS = LIB$ANALYZE_SDESC_R2 ( .STRNG_DESC_ADDR ;
307 0583 2 STRING_LEN, STRING_ADDR ) ;
308 0584 2
309 0585 2 IF NOT .RET_STATUS THEN RETURN (.RET_STATUS) ;
310 0586 2 END
311 0587 2
312 0588 2 ELSE ! Fetch length and address directly
313 0589 2
314 0590 2 BEGIN
315 0591 2 STRING_LEN = .STRNG_DESC_ADDR [DSC$W_LENGTH] ;
316 0592 2 STRING_ADDR = .STRNG_DESC_ADDR [DSC$A_POINTER] ;
317 0593 2 END;
318 0594 2
319 0595 2 !+
320 0596 2 !- Scan the keyword table for a match with the caller's string.
321 0597 2
322 0598 2
323 0599 2 INCR INDEX FROM 1 TO .KEY_TABLE BY 2 DO ! Loop until the end of
324 0600 2 ! the keyword table.
325 0601 2 BEGIN
326 0602 2
327 0603 2 !+
328 0604 2 !- First make sure that the caller's string is not longer than
329 0605 2 the current keyword.
330 0606 2
331 0607 2 IF (.STRING_LEN LEQU $LIB_KEY_LENGTH (KEY_TABLE [.INDEX]))
```

```
332 0608
333 0609
334 0610
335 0611
336 0612
337 0613
338 0614
339 0615
340 0616
341 0617
342 0618
343 0619
344 0620
345 0621
346 0622
347 0623
348 0624
349 0625
350 0626
351 0627
352 0628
353 0629
354 0630
355 0631
356 0632
357 0633
358 0634
359 0635
360 0636
361 0637
362 0638
363 0639
364 0640
365 0641
366 0642
367 0643
368 0644
369 0645
370 0646
371 0647
372 0648
373 0649
374 0650
375 0651
376 0652
377 0653
378 0654
379 0655
380 0656
381 0657
382 0658
383 0659
384 0660
385 0661
386 0662
387 0663
388 0664

THEN
  !+
  ! If the caller's string matches the current keyword begin
  ! additional checking.
  IF CH$EQL(.STRING_LEN, .STRING_ADDR,
    .STRING_LEN, $LIB_KEY_STRING (KEY_TABLE [.INDEX]))
  THEN
    BEGIN
      !+
      ! If the caller's search string is the same length as
      ! the keyword it matches
      IF (.STRING_LEN EQLU
        $LIB_KEY_LENGTH (KEY_TABLE [.INDEX]))
      THEN
        BEGIN ! special exact-match processing.
          MATCH = .INDEX; ! Save the current keyword table
                        ! index,
          RETURN_CODE = SS$_NORMAL; ! indicate a keyword
                        ! match was found,
          EXITLOOP; ! and bypass further key
                        ! word table scanning.
          END; ! special exact-match processing

        !+
        ! If a match has not already been found,
        IF (.RETURN_CODE EQL LIB$_UNRKEY)
        THEN
          BEGIN
            MATCH = .INDEX; ! save the current keyword table
                        ! index
            RETURN_CODE = SS$_NORMAL; ! and indicate a match
                        ! has been found.
          END

        ELSE
          ! Otherwise, indicate that a multiple
          ! keyword match has been found

          BEGIN
            RETURN_CODE = LIB$_AMBKEY; ! (i.e., no enough
                        ! characters provided)
            EXITLOOP; ! and exit the keyword
                        ! scanning loop.
          END;

        END; ! End of keyword match processing.
      END; ! End of the keyword table searching loop.

    !+
    ! If a keyword match was found, return the keyword information to the
    ! caller.
  !-
```



```
389 0665 3
390 0666 3
391 0667 3
392 0668 3
393 0669 3
394 0670 3
395 0671 3
396 0672 3
397 0673 3
398 0674 3
399 0675 3
400 0676 3
401 0677 3
402 0678 3
403 0679 3
404 0680 3
405 0681 3
406 0682 4
407 0683 3
408 0684 4
409 0685 4
410 0686 4
411 0687 4
412 0688 4
413 0689 4
414 0690 4
415 0691 4
416 0692 4
417 0693 4
418 0694 4
419 0695 4
420 0696 4
421 0697 4
422 0698 4
423 0699 4
424 0700 4
425 0701 4
426 0702 4
427 0703 4
428 0704 4
429 0705 4
430 0706 4
431 0707 4
432 0708 5
433 0709 4
434 0710 5
435 0711 5
436 0712 5
437 0713 5
438 0714 5
439 0715 5
440 0716 6
441 0717 6
442 0718 6
443 0719 6
444 0720 6
445 0721 5
```

```
IF (.RETURN_CODE NEQ LIB$_UNRKEY) ! Make sure a keyword match was
! found.
THEN
  BEGIN ! match was found code
    !+
    ! If the caller provided a parameter to receive the key value,
    ! return it to him.
    !-

    IF ( NOT NULLPARAMETER (3)) THEN
      KEY_VALUE = $LIB_KEY_VALUE (KEY_TABLE [.MATCH]);

    !+
    ! If the caller has provided a descriptor to receive the matched
    ! key, return it to him.
    !-

    IF ( NOT NULLPARAMETER (4))
    THEN
      BEGIN ! returning optional arguments

        LOCAL
          RET_STATUS;

        RET_STATUS = LIB$SCOPY_R DX6 (
          $LIB_KEY_LENGTH (KEY_TABLE [.MATCH]),
          $LIB_KEY_STRING (KEY_TABLE [.MATCH]),
          .FULL_DESC_ADDR ) ;

        !+
        ! If the copy failed, record the status
        !-
        IF NOT .RET_STATUS THEN RETURN_CODE = .RET_STATUS ;

        !+
        ! If the caller is using fixed length strings, he may want
        ! to know how many characters we actually stored, not
        ! counting trailing pads.
        ! There is no need to check status on call to
        ! LIB$ANALYZE_SDESC_R2.
        ! If the descriptor was bad, it would have gotten caught
        ! during the copy operation above.
        !-

        IF ( NOT NULLPARAMETER (5))
        THEN
          BEGIN ! returning length
            LOCAL
              FULL_LEN,
              FULL_ADDR;
            IF .FULL_DESC_ADDR [DSC$B_CLASS] GTRU DSC$K_CLASS_D
            THEN ! Use general length extraction
              BEGIN
                LIB$ANALYZE_SDESC_R2 ( .FULL_DESC_ADDR ;
                  FULL_LEN, FULL_ADDR ) ;
              END
            ELSE
              ! Fetch length directly
```



```

: 446      0722      5      FULL_LEN = .FULL_DESC_ADDR [DSC$W_LENGTH] ;
: 447      0723
: 448      0724      5      OUTLEN [0] = MIN ($LIB_KEY_LENGTH (KEY_TABLE [.MATCH]),
: 449      0725      5      .FULL_LEN);
: 450      0726      4      END;      ! returning length
: 451      0727      4
: 452      0728      3      END;      ! returning optional arguments
: 453      0729      3
: 454      0730      2      END;      ! match as found code
: 455      0731      2
: 456      0732      2      !+
: 457      0733      2      Return the current keyword match return code to the caller.
: 458      0734      2      -
: 459      0735      2      RETURN (.RETURN_CODE);      ! Return the current match code to the
: 460      0736      2      caller.
: 461      0737      1      END;      ! End of LIB$LOOKUP_KEY routine
```

```
.TITLE LIB$LOOKUP_KEY
.IDENT \1-012\
```

```
.EXTRN LIB$ANALYZE_SDESC_R2
.EXTRN LIB$COPY_R_DX6
.EXTRN LIB$STRTRU, LIB$_AMBKEY
.EXTRN LIB$_UNRKEY, LIB$_INVARG
```

```
.PSECT _LIB$CODE, NOWRT, SHR, PIC, 2
```

```
.ENTRY LIB$LOOKUP_KEY, Save R2,R3,R4,R5,R6,R7,R8,- : 0437
R9,R10,R11
```

```
MOVAB LIB$ANALYZE_SDESC_R2, R11
MOVL #LIB$_UNRKEY, R10
CMPB (AP), #2 : 0565
```

```
BGEQU 1$
MOVL #LIB$_INVARG, R0 : 0567
```

```
RET
MOVL R10, RETURN_CODE : 0572
```

```
MOVL STRNG_DESC_ADDR, R3 : 0578
```

```
CMPB 3(R3), #2
BLEQU 2$ : 0582
```

```
MOVL R3, R0
JSB LIB$ANALYZE_SDESC_R2
```

```
MOVQ R1, R7
BLBS RET_STATUS, 3$ : 0585
```

```
RET
MOVZWL (R3), STRING_LEN : 0591
```

```
MOVL 4(R3), STRING_ADDR : 0592
```

```
MNEGL #1, INDEX : 0607
```

```
BRB 7$
MOVL @KEY_TABLE_ADDR[INDEX], R5
```

```
CMPZV #0, #8, (R5), STRING_LEN
```

```
BLSSU 7$
CMPC3 STRING_LEN, (STRING_ADDR), 1(R5) : 0613
```

```
BNEQ 7$
CMPZV #0, #8, (R5), STRING_LEN : 0623
```

```
BNEQ 5$
MOVL INDEX, MATCH : 0626
```

57 65

01 A5

57 65

```

OFFC 00000
5B 00000000G 00 9E 00002
5A 00000000G 8F D0 00009
02 6C 91 00010
08 1E 00013
50 00000000G 8F D0 00015
04 0001C
59 5A D0 0001D 1$:
53 04 AC D0 00020
02 03 A3 91 00024
0C 1B 00028
50 53 D0 0002A
6B 16 0002D
57 51 7D 0002F
08 50 E8 00032
04 00035
57 63 3C 00036 2$:
58 04 A3 D0 00039
54 01 CE 0003D 3$:
38 11 00040
55 08 BC 44 D0 00042 4$:
08 00 ED 00047
2C 1F 0004C
68 57 29 0004E
25 12 00053
08 00 ED 00055
08 12 0005A
56 54 D0 0005C
```


59	01	D0	0005F	MOVL	#1, RETURN_CODE	: 0628
	1D	11	00062	BRB	8\$	: 0625
5A	59	D1	00064	5\$: CMPL	RETURN_CODE, R10	: 0637
	08	12	00067	BNEQ	6\$	
56	54	D0	00069	MOVL	INDEX, MATCH	: 0640
59	01	DC	0006C	MOVL	#1, RETURN_CODE	: 0642
	09	11	0006F	BRB	7\$	: 0637
59	00000000G	8F	D0	6\$: MOVL	#LIB\$_AMBKEY, RETURN_CODE	: 0650
		07	11	BRB	8\$	: 0649
FFC1	54	02	08	BC	F1 0007A	7\$: ACBL
		5A		59	D1 00081	8\$: CMPL
				66	13 00084	BEQL
		03		6C	91 00086	CMPB
				0F	1F 00089	BLSSU
			0C	AC	D5 0008B	TSTL
				0A	13 0008E	BEQL
50	08	BC	46	DE	00090	MOVAL
OC	BC	04	A0	D0	00095	MOVL
	04		6C	91	0009A	9\$: CMPB
			4D	1F	0009D	BLSSU
		10	AC	D5	0009F	TSTL
			48	13	000A2	BEQL
57	10	AC	D0	000A4	MOVL	FULL_DESC_ADDR, R7
58	08	BC	46	D0	000A8	MOVL
51	01	A8	9E	000AD	MOVAB	1(R8), R1
52		57	D0	000B1	MOVL	R7, R2
50		68	9A	000B4	MOVZBL	(R8), R0
	00000000G	00	16	000B7	JSB	LIB\$SCOPY_R_DX6
03		50	E8	000BD	BLBS	RET_STATUS, -10\$
59		50	D0	000C0	MOVL	RET_STATUS, RETURN_CODE
05		6C	91	000C3	10\$: CMPB	(AP), #5
		24	1F	000C6	BLSSU	14\$
		14	AC	D5	000C8	TSTL
			1F	13	000CB	BEQL
02	03	A7	91	000CD	CMPB	3(R7), #2
		07	1B	000D1	BLEQU	11\$
50		57	D0	000D3	MOVL	R7, R0
		6B	16	000D6	JSB	LIB\$ANALYZE_SDESC_R2
		03	11	000D8	BRB	12\$
51		67	3C	000DA	11\$: MOVZWL	(R7), FULL_LEN
50		68	9A	000DD	12\$: MOVZBL	(R8), R0
51		50	D1	000E0	CMPB	R0, FULL_LEN
		03	15	000E3	BLEQ	13\$
50		51	D0	000E5	MOVL	FULL_LEN, R0
14	BC	50	B0	000E8	13\$: MOVW	R0, \$OUTLEN
	50	59	D0	000EC	14\$: MOVL	RETURN_CODE, R0
			04	000EF	RET	: 0737

; Routine Size: 240 bytes, Routine Base: _LIB\$CODE + 0000

: 462 0738 1
: 463 0739 1 END
: 464 0740 1
: 465 0741 0 ELUDOM

! End of LIB\$LOOKUP_KEY module

PSECT SUMMARY

```
:
:      Name                Bytes                Attributes
:  _LIB$CODE              240  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
```

Library Statistics

```
:
:      File                ----- Symbols ----- Pages      Processing
:                        Total   Loaded   Percent   Mapped      Time
:  _$255$DUA28:[SYSLIB]STARLET.L32;1    9776         5         0       581      00:00.7
```

COMMAND QUALIFIERS

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

```
: Size:                240 code + 0 data bytes
: Run Time:            00:06.1
: Elapsed Time:        00:36.4
: Lines/CPU Min:       7252
: Lexemes/CPU-Min: 25321
: Memory Used: 114 pages
: Compilation Complete
```


0208 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

