


```

LL      IIIIII  BBBB BBBB  CCCCCCCC  000000  MM      MM      MM      MM      000000  NN      NN
LL      IIIIII  BBBB BBBB  CCCCCCCC  000000  MM      MM      MM      MM      000000  NN      NN
LL      II      BB      BB  CC      CC  00      00  MMMM  MMMM  MMMM  MMMM  00      00  NN      NN
LL      II      BB      BB  CC      CC  00      00  MMMM  MMMM  MMMM  MMMM  00      00  NN      NN
LL      II      BB      BB  CC      CC  00      00  MM  MM  MM  MM  MM  MM  00      00  NNNN  NN
LL      II      BB      BB  CC      CC  00      00  MM  MM  MM  MM  MM  MM  00      00  NNNN  NN
LL      II      BBBB BBBB  CC      CC  00      00  MM      MM  MM  MM  MM  MM  00      00  NN  NN
LL      II      BBBB BBBB  CC      CC  00      00  MM      MM  MM  MM  MM  MM  00      00  NN  NN
LL      II      BB      BB  CC      CC  00      00  MM      MM  MM  MM  MM  MM  00      00  NN  NN
LL      II      BB      BB  CC      CC  00      00  MM      MM  MM  MM  MM  MM  00      00  NN  NN
LL      II      BB      BB  CC      CC  00      00  MM      MM  MM  MM  MM  MM  00      00  NN  NN
LL      II      BB      BB  CC      CC  00      00  MM      MM  MM  MM  MM  MM  00      00  NN  NN
LLLLLLLL  IIIIII  BBBB BBBB  CCCCCCCC  000000  MM      MM  MM  MM  MM      000000  NN      NN
LLLLLLLL  IIIIII  BBBB BBBB  CCCCCCCC  000000  MM      MM  MM  MM  MM      000000  NN      NN

```



```

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 LIB$COMMON ( ! Get and put common
2 0002 0
3 0003 0 IDENT = '1-006' ! File: LIBCOMMON.B32 Edit: RKR1006
4 0004 0
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY: VAX-11 COMMON RTL
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module contains routines to move strings to and from the
37 0037 1 process permanent common area.
38 0038 1
39 0039 1 ENVIRONMENT: Native mode VAX processor, Any mode.
40 0040 1
41 0041 1 AUTHOR: V. Eriksson, CREATION DATE: 7-Nov-79
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 VERSION 1.0
46 0046 1
47 0047 1 01 -
48 0048 1 1-002 - Fit into the RTL's build procedure and convert to LIB$ width,
49 0049 1 case and commenting conventions. JBS 02-DEC-1979
50 0050 1 1-003 - Change return codes from STR$ to LIB$. JBS 22-JAN-1980
51 0051 1 1-004 - Change copy call from STR$COPY DX to LIB$SCOPY_DXDX6. This
52 0052 1 eliminates the need for a handler and the need to convert
53 0053 1 STR$ statuses to LIB$ statuses. Also faster.
54 0054 1 RKR 26-MAY-1981.
55 0055 1 1-005 - Add special-case code to process classes of string descriptors
56 0056 1 that "read" like fixed strings. RKR 7-OCT-1981.
57 0057 1 1-006 - Redirect jsb's from LIB$ANALYZE_DESC_R3 to

```


LIB\$COMMON
1-006

C 3
16-Sep-1984 00:39:37 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:38:27 [LIBRTL.SRC]LIBCOMMON.B32;1

Page 2
(1)

```
: 58      0058 1 | LIB$ANALYZE_SDESC_R2. Use LIB$SCOPY_R_DX6 instead of
: 59      0059 1 | LIB$SCOPY_DXDX6 to pick up some more speed. RKR 18-NOV-1981.
: 60      0060 1 | --
: 61      0061 1 |
: 62      0062 1 | <blf/page>
```

```

64      0063 1 |
65      0064 1 | SWITCHES :
66      0065 1 |
67      0066 1 |
68      0067 1 | SWITCHES ADDRESSING MODE
69      0068 1 |             (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
70      0069 1 |
71      0070 1 |
72      0071 1 | LINKAGES
73      0072 1 |
74      0073 1 |
75      0074 1 | REQUIRE 'RTLIN:STRLNK';           ! Linkages for LIB$ANALYZE_SDESC_R2
76      0259 1 |
77      0260 1 |
78      0261 1 | TABLE OF CONTENTS:
79      0262 1 |
80      0263 1 |
81      0264 1 | FORWARD ROUTINE
82      0265 1 |     LIB$PUT_COMMON,           ! Puts a string into the common area
83      0266 1 |     LIB$GET_COMMON;          ! Gets a string from the common area
84      0267 1 |
85      0268 1 |
86      0269 1 | INCLUDE FILES:
87      0270 1 |
88      0271 1 |
89      0272 1 | LIBRARY 'RTLSTARLE';           ! Get descriptor field definitions
90      0273 1 |
91      0274 1 | REQUIRE 'RTLIN:RTLPSECT';       ! Define the declare_psects macro
92      0369 1 |
93      0370 1 |
94      0371 1 | MACROS:
95      0372 1 |
96      0373 1 |     NONE
97      0374 1 |
98      0375 1 | EQUATED SYMBOLS:
99      0376 1 |
100     0377 1 |
101     0378 1 | LITERAL
102     0379 1 |     COMMON_LENGTH = 256,       ! Length of common area in P1 space
103     0380 1 |     COMMON_OFFSET = 0;        ! Offset of common area from base in
104     0381 1 |                                ! P1 space pointed to by descriptor
105     0382 1 |
106     0383 1 |
107     0384 1 | PSECTS:
108     0385 1 |
109     0386 1 | DECLARE_PSECTS (LIB);         ! Declare psects for the LIB$ facility
110     0387 1 |
111     0388 1 | OWN STORAGE:
112     0389 1 |
113     0390 1 |     NONE
114     0391 1 |
115     0392 1 | EXTERNAL REFERENCES:
116     0393 1 |
117     0394 1 |
118     0395 1 | EXTERNAL ROUTINE
119     0396 1 |     LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK, ! Routine to
120     0397 1 |                                ! extract length
```


LIB\$COMMON
1-006

E 3
16-Sep-1984 00:39:37 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:38:27 [LIBRTL.SRC]LIBCOMMON.B32;1

Page 4
(2)

```
: 121      0398 1
: 122      0399 1
: 123      0400 1      LIB$COPY_R_DX6 : STRING_JSB;      ! and address of
: 124      0401 1      ! 1st data byte
: 125      0402 1      ! Routine to copy string
: 126      0403 1      EXTERNAL
: 127      0404 1      SYSSGL_COMMON : REF BLOCK [, BYTE];      ! Pointer to descriptor
: 128      0405 1      ! of common area in P1
: 129      0406 1      ! space.
: 130      0407 1      !+
: 131      0408 1      ! The following error codes are used in this module
: 132      0409 1      !-
: 133      0410 1
: 134      0411 1      EXTERNAL LITERAL
: 135      0412 1      LIB$_INVSTRDES,      ! invalid string descr.
: 136      0413 1      LIB$_STRTRU;      ! string truncated
: 137      0414 1
```

LIB
1-0

Page 5
(3)

```

139      0415 1 GLOBAL ROUTINE LIB$PUT_COMMON (      ! Put string into common area
140      0416 1
141      0417 1      STRING_DESCR,      ! String to put
142      0418 1      BYTES_COPIED      ! Number of bytes copied
143      0419 1
144      0420 1      ) =
145      0421 1
146      0422 1  !++
147      0423 1  FUNCTIONAL DESCRIPTION:
148      0424 1
149      0425 1      The string (described by first parameter) is copied to the
150      0426 1      common area. The length of the string is moved to the
151      0427 1      first longword of the area. If the string is too long, then
152      0428 1      only part of the string is copied. The number of bytes copied is
153      0429 1      returned in the parameter bytes_copied, if the routine is called
154      0430 1      by two parameters.
155      0431 1
156      0432 1  FORMAL PARAMETERS:
157      0433 1
158      0434 1      string_descr.rt.dx -      Pointer to descriptor of string to be
159      0435 1      copied
160      0436 1      bytes_copied.wv.r -      Pointer to word to receive # bytes
161      0437 1      copied
162      0438 1      (optional parameter)
163      0439 1
164      0440 1  IMPLICIT INPUTS:
165      0441 1
166      0442 1      SYS$GL_COMMON      -      Pointer to descriptor of the common area
167      0443 1
168      0444 1  IMPLICIT OUTPUTS:
169      0445 1
170      0446 1      The common area is (partly) filled by the string.
171      0447 1      The length of the string is put in the first longword
172      0448 1      of the common area.
173      0449 1
174      0450 1  CONDITION CODES:
175      0451 1
176      0452 1      SSS_NORMAL      - success
177      0453 1      LIB$INVSTRDES - invalid string descriptor
178      0454 1      LIB$STRTRU  - common had room for only part of the string
179      0455 1
180      0456 1  SIDE EFFECTS:
181      0457 1
182      0458 1      NONE
183      0459 1
184      0460 1  --
185      0461 1
186      0462 2      BEGIN
187      0463 2
188      0464 2      BUILTIN
189      0465 2      NULLPARAMETER;
190      0466 2
191      0467 2      MAP
192      0468 2      STRING_DESCR : REF BLOCK [, BYTE],      ! Points to the string
193      0469 2      BYTES_COPIED : REF VECTOR [, WORD];      ! Points to word to
194      0470 2      ! receive number of
195      0471 2

```



```

: 196      0472 2
: 197      0473 2
: 198      0474 2
: 199      0475 2
: 200      0476 2
: 201      0477 2
: 202      0478 2
: 203      0479 2
: 204      0480 2
: 205      0481 2
: 206      0482 2
: 207      0483 2
: 208      0484 2
: 209      0485 2
: 210      0486 2
: 211      0487 2
: 212      0488 2
: 213      0489 2
: 214      0490 2
: 215      0491 2
: 216      0492 2
: 217      0493 2
: 218      0494 2
: 219      0495 2
: 220      0496 2
: 221      0497 2
: 222      0498 2
: 223      0499 3
: 224      0500 3
: 225      0501 3
: 226      0502 2
: 227      0503 2
: 228      0504 3
: 229      0505 3
: 230      0506 3
: 231      0507 3
: 232      0508 2
: 233      0509 2
: 234      0510 2
: 235      0511 2
: 236      0512 2
: 237      0513 2
: 238      0514 2
: 239      0515 2
: 240      0516 3
: 241      0517 3
: 242      0518 3
: 243      0519 2
: 244      0520 2
: 245      0521 2
: 246      0522 2
: 247      0523 2
: 248      0524 2
: 249      0525 2
: 250      0526 2
: 251      0527 2
: 252      0528 2

LOCAL
SRC_LEN,
SRC_ADDR,

COMMON_DESCR : BLOCK [8, BYTE],

STS;

! bytes copied

! Length of source
! Address of 1st data
! byte of source
! Descriptor of common
! string area
! Status to return to
! caller

+
Initialise the common string area descriptor.
-
COMMON_DESCR [DSC$W_LENGTH] =
MINU (.SYS$GL_COMMON [DSC$W_LENGTH], COMMON_LENGTH) - 4;
COMMON_DESCR [DSC$A_POINTER] =
.SYS$GL_COMMON [DSC$A_POINTER] + COMMON_OFFSET + 4;
COMMON_DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_2;
COMMON_DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;

+
Extract length and address of 1st data byte from source string
-
IF .STRING_DESCR [DSC$B_CLASS] GTRU DSC$K_CLASS_D
THEN
! generalized extraction needed
BEGIN
STS = LIB$ANALYZE_SDESC_R2 ( .STRING_DESCR ;
SRC_LEN, SRC_ADDR ) ;
END
ELSE
! can fetch pieces directly
BEGIN
SRC_LEN = .STRING_DESCR [DSC$W_LENGTH] ;
SRC_ADDR = .STRING_DESCR [DSC$A_POINTER] ;
STS = SS$_NORMAL ;
END ;

+
If status returned was not a success, skip the copy, but update
BYTES_COPIED (if present).
-
IF NOT .STS
THEN
BEGIN ! bad source descriptor
IF NOT NULLPARAMETER (2) THEN BYTES_COPIED [0] = 0 ;
RETURN .STS ;
END ; ! bad source descriptor

+
Move as much as possible of the string to the common string area.
-
STS = LIB$SCOPY_R_DX6 ( .SRC_LEN, .SRC_ADDR, COMMON_DESCR ) ;

+
Put the number of bytes copied in the first word of the common
area.
```



```

: 253      0529 2  !-
: 254      0530 2  .SYS$GL_COMMON [DSC$A_POINTER] =
: 255      0531 2  MINU (.SRC_LEN, .COMMON_DESCR [DSC$W_LENGTH] );
: 256      0532 2
: 257      0533 2  !+
: 258      0534 2  Put number of bytes copied in the parameter bytes_copied (if the
: 259      0535 2  second parameter is specified).
: 260      0536 2  !-
: 261      0537 2  IF NOT NULLPARAMETER (2) THEN
: 262      0538 2  BYTES_COPIED [0] = .(.SYS$GL_COMMON [DSC$A_POINTER]);
: 263      0539 2
: 264      0540 2  RETURN .STS;
: 265      0541 1  END;
```

! End of LIB\$PUT_COMMON

```
.TITLE LIB$COMMON
.IDENT \1-006\
```

```
.EXTRN LIB$ANALYZE_SDESC_R2
.EXTRN LIB$SCOPY_R_DX6
.EXTRN SYS$GL_COMMON, LIB$_INVSTRDES
.EXTRN LIB$_STRTRU
```

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

```
.ENTRY LIB$PUT_COMMON, Save R2,R3,R4,R5,R6,R7,R8
```

```

MOVAB SYS$GL_COMMON, R8 ; 0415
SUBL2 #8, SP
MOVL SYS$GL_COMMON, R1 ; 0487
MOVZWL (R1), R0
CMPW R0, #256
BLEQU 1$
MOVZWL #256, R0
SUBW3 #4, R0, COMMON_DESCR
ADDL3 #4, 4(R1), COMMON_DESCR+4 ; 0489
MOVW #256, COMMON_DESCR+2 ; 0490
MOVL STRING_DESCR, R3 ; 0496
CMPB 3(R3), #2
BLEQU 2$
MOVL R3, R0 ; 0499
JSB LIB$ANALYZE_SDESC_R2
MOVL R0, STS
MOVL R1, R7
MOVL R2, R5
BRB 3$ ; 0496
MOVZWL (R3), SRC_LEN ; 0505
MOVL 4(R3), SRC_ADDR ; 0506
MOVL #1, STS ; 0507
BLBS STS, 4$ ; 0514
CMPB (AP), #2 ; 0517
BLSSU 6$
TSTL 8(AP)
BEQL 6$
CLRW @BYTES_COPIED
BRB 6$ ; 0518
MOVAB COMMON_DESCR, R2 ; 0524
MOVL SRC_ADDR, R1
```

LIB\$COMMON
1-006

I 3
16-Sep-1984 00:39:37
14-Sep-1984 12:38:27

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]LIBCOMMON.B32;1

Page 8
(3)

LIB
1-0

51	6E	50	00000000G	57	D0	0006E	MOVL	SRC_LEN, R0	:	
		54		00	16	00071	JSB	LIB\$COPY_R_DX6	:	
		50		50	D0	00077	MOVL	R0, STS	:	0530
		51		68	D0	0007A	MOVL	SY\$SGL_COMMON, R0	:	0531
		10		57	D0	0007D	MOVL	SRC_LEN, R1	:	
				00	ED	00080	CMPZV	#0, #16, COMMON_DESCR, R1	:	
				03	1E	00085	BGEQU	5\$	:	
		51		6E	3C	00087	MOVZWL	COMMON_DESCR, R1	:	
04		B0		51	D0	0008A	MOVL	R1, @4(R0)	:	
		02		6C	91	0008E	CMPB	(AP), #2	:	0537
				0A	1F	00091	BLSSU	6\$	:	
			08	AC	D5	00093	TSTL	8(AP)	:	
				05	13	00096	BEQL	6\$	:	
08	BC		04	B0	B0	00098	MOVW	@4(R0), @BYTES_COPIED	:	0538
	50			54	D0	0009D	MOVL	STS, R0	:	0540
				04	00	000A0	RET		:	0541

; Routine Size: 161 bytes, Routine Base: _LIB\$CODE + 0000

; 266 0542 1


```

268 0543 1 GLOBAL ROUTINE LIB$GET_COMMON ( ! Get a string from common area
269 0544 1
270 0545 1 BUFFER_DESCR, ! Where to put string
271 0546 1 BYTES_COPIED ! Number of bytes actually copied
272 0547 1
273 0548 1 ) =
274 0549 1
275 0550 1 ++
276 0551 1 FUNCTIONAL DESCRIPTION:
277 0552 1
278 0553 1 The string in the common area is copied to the
279 0554 1 buffer given by the parameter buffer_descr. The string
280 0555 1 length is fetched from the first longword of the common
281 0556 1 area. If the string is too long for the buffer, then only
282 0557 1 part of the string is copied. The number of bytes copied
283 0558 1 is returned by the parameter bytes_copied, if the routine is
284 0559 1 called by two parameters.
285 0560 1
286 0561 1 FORMAL PARAMETERS:
287 0562 1
288 0563 1 buffer_descr.rz.dx - Pointer to descr of buffer to receive
289 0564 1 string
290 0565 1 bytes_copied.wv.r - Pointer to word to receive # bytes
291 0566 1 copied
292 0567 1 (optional parameter)
293 0568 1
294 0569 1 IMPLICIT INPUTS:
295 0570 1
296 0571 1 SYSS$GL_COMMON - Pointer to descriptor of common area
297 0572 1
298 0573 1 IMPLICIT OUTPUTS:
299 0574 1
300 0575 1 The buffer described by the parameter buffer_descr is (partly)
301 0576 1 filled by the string from the common area.
302 0577 1
303 0578 1 ROUTINE VALUE:
304 0579 1
305 0580 1 SSS$NORMAL - success
306 0581 1 LIB$INVSTRDES - invalid string descriptor
307 0582 1 LIB$STRTRU - string was longer than buffer and was
308 0583 1 truncated
309 0584 1
310 0585 1 SIDE EFFECTS:
311 0586 1
312 0587 1 NONE
313 0588 1
314 0589 1 --
315 0590 1
316 0591 2 BEGIN
317 0592 2
318 0593 2 BUILTIN
319 0594 2 NULLPARAMETER;
320 0595 2
321 0596 2 MAP
322 0597 2 BUFFER_DESCR : REF BLOCK [ , BYTE], ! Points to receive buffer
323 0598 2 BYTES_COPIED : REF VECTOR [ , WORD], ! Points to word to receive
324 0599 2 ! number of bytes copied

```

```

325 0600 2
326 0601 2
327 0602 2
328 0603 2
329 0604 2
330 0605 2
331 0606 2
332 0607 2
333 0608 2
334 0609 2
335 0610 2
336 0611 2
337 0612 2
338 0613 2
339 0614 2
340 0615 2
341 0616 2
342 0617 2
343 0618 2
344 0619 2
345 0620 2
346 0621 2
347 0622 2
348 0623 2
349 0624 2
350 0625 2
351 0626 2
352 0627 2
353 0628 2
354 0629 2
355 0630 2
356 0631 2
357 0632 2
358 0633 2
359 0634 2
360 0635 2
361 0636 2
362 0637 2
363 0638 2
364 0639 2
365 0640 2
366 0641 2
367 0642 2
368 0643 2
369 0644 2
370 0645 2
371 0646 1

LOCAL
    IN_LEN,      ! Size of input area
    OUT_LEN,     ! Size of output area
    OUT_ADDR,    ! Address of 1st data byte of
                  ! output area
    COMMON_DESCR : BLOCK [8, BYTE], ! Descriptor of the common
                  ! string area.
    STS;         ! Status to return to caller

!+ Move as much as possible of the string from the common area to
!- the buffer.
    IN_LEN = MINU ( (.SY$SGL_COMMON [DSC$A_POINTER]),
                  COMMON_LENGTH - 4 );

    STS = LIB$SCOPY_R_DX6 ( .IN_LEN,      ! length
                          .SY$SGL_COMMON [DSC$A_POINTER] + ! addr
                          COMMON_OFFSET + 4,
                          .BUFFER_DESCR );

! Move the number of bytes copied to the parameter bytes_copied, if the
! second parameter is specified.
    IF NOT NULLPARAMETER (2)
    THEN
        BEGIN
            !+ Determine how many bytes actually got copied into destination
            !- buffer by LIB$SCOPY_R_DX6.
            IF .BUFFER_DESCR [DSC$B_CLASS] GTRU DSC$K_CLASS_D
            THEN
                ! generalized extraction of length
                BEGIN
                    LIB$ANALYZE_SDESC_R2 ( .BUFFER_DESCR ; OUT_LEN, OUT_ADDR );
                END
            ELSE
                ! can fetch length directly
                OUT_LEN = .BUFFER_DESCR [DSC$W_LENGTH] ;

                BYTES_COPIED [0] = MINU (.IN_LEN, .OUT_LEN);
            END;
        END;
    RETURN .STS;
END;
! End of LIB$GET_COMMON
```

```

000000FC  SE 00000000G 04 01FC 0000
          51 00000000G 08 C2 00002
          50 00000000G 00 D0 00005
          8F 00000000G B1 D0 0000C
          50 00000000G 50 D1 00010
```

```

.ENTRY LIB$GET_COMMON, Save R2,R3,R4,R5,R6,R7,R8 : 0543
SUBL2 #8, SP :
MOVL SY$SGL_COMMON, R1 : 0614
MOVL @4(R1), R0 : 0615
CMPL R0, #252 :
```


			04	1B	00017	BLEQU	1\$	
	50	FC	8F	9A	00019	MOVZBL	#252, R0	
	58		50	D0	0001D	1\$:	MOV	IN_LEN
	57	04	AC	D0	00020	MOV	BUFFER_DESCR, R7	0614
51	A1		04	C1	00024	ADDL3	#4, 4(R1), R1	0620
	52		57	D0	00029	MOV	R7, R2	0619
	50		58	D0	0002C	MOV	IN_LEN, R0	0617
		00000000G	00	16	0002F	JSB	LIB\$SCOPY_R_DX6	
	53		50	D0	00035	MOV	R0, STS	
	02		6C	91	00038	CMPB	(AP), #2	0625
			28	1F	0003B	BLSSU	5\$	
		08	AC	D5	0003D	TSTL	8(AP)	
			23	13	00040	BEQ	5\$	
	02	03	A7	91	00042	CMPB	3(R7), #2	0632
			0B	1B	00046	BLEQU	2\$	
	50		57	D0	00048	MOV	R7, R0	0635
		00000000G	00	16	0004B	JSB	LIB\$ANALYZE_SDESC_R2	
			03	11	00051	BRB	3\$	0632
	51		67	3C	00053	2\$:	MOVZWL	(R7), OUT_LEN
	50		58	D0	00056	3\$:	MOV	IN_LEN, R0
	51		50	D1	00059	CMP	R0, OUT_LEN	0640
			03	1B	0005C	BLEQU	4\$	0642
	50		51	D0	0005E	MOV	OUT_LEN, R0	
	08	BC	50	B0	00061	4\$:	MOVW	R0, @BYTES_COPIED
	50		53	D0	00065	5\$:	MOV	STS, R0
			04	00068	RET			0645
								0646

: Routine Size: 105 bytes, Routine Base: _LIB\$CODE + 00A1

: 372	0647	1	
: 373	0648	1	END
: 374	0649	1	
: 375	0650	0	ELUDOM

! End of module LIB\$COMMON

PSECT SUMMARY

Name	Bytes	Attributes
_LIB\$CODE	266	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	8	0	581	00:00.8

LIB\$COMMON
1-006

M 3
16-Sep-1984 00:39:37
14-Sep-1984 12:38:27

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]LIBCOMMON.B32;1

Page 12
(4)

**F

COMMAND QUALIFIERS

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

: Size: 266 code + 0 data bytes
: Run Time: 00:05.8
: Elapsed Time: 00:21.7
: Lines/CPU Min: 6712
: Lexemes/CPU-Min: 26426
: Memory Used: 75 pages
: Compilation Complete

0204 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

