

STB
1-0

.....

```

SSSSSSSSS  TTTTTTTTT  RRRRRRRR  TTTTTTTTT  RRRRRRRR  AAAAAA  NN  NN  SSSSSSSS  LL
SSSSSSSSS  TTTTTTTTT  RRRRRRRR  TTTTTTTTT  RRRRRRRR  AAAAAA  NN  NN  SSSSSSSS  LL
SS      TT      RR      TT      RR      AA      NN  NN  SS      LL
SS      TT      RR      TT      RR      AA      NN  NN  SS      LL
SS      TT      RR      TT      RR      AA      NNNN  NN  SS      LL
      SSSSSS  RRRRRRRR  TT      RRRRRRRR  AA      AA      NNNN  NN  SSSSSS  LL
      SSSSSS  RRRRRRRR  TT      RRRRRRRR  AA      AA      NN  NN  SSSSSS  LL
      SS      RR      TT      RR      RRRRRRRR  AA      AA      NN  NN  SS      LL
      SS      RR      TT      RR      RR      AA      AA      NN  NN  SS      LL
      SS      RR      TT      RR      RR      AA      AA      NN  NN  SS      LL
SSSSSSSSS  TT      RR      TT      RR      RR      AA      AA      NN  NN  SSSSSSSS  LL
SSSSSSSSS  TT      RR      TT      RR      RR      AA      AA      NN  NN  SSSSSSSS  LL

```

.....

.....

.....

.....

```

LL      SSSSSSSS  SSSSSSSS
LL      SSSSSSSS
LL      SS
LL      SS
LL      SS
LL      SS
LL      SSSSSS
LL      SSSSSS
LL      SS
LL      SS
LL      SS
LL      SS
LL      SSSSSS
LLLLLLLLLL  SSSSSSSS
LLLLLLLLLL  SSSSSSSS

```



```
1 0001 0 MODULE STR$TRANSLATE (      ! copy source to dest and translate
2 0002 0
3 0003 0 IDENT = '1-008' ! File: STRTRANSL.B32 Edit: PDG1008
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: String support library
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This module takes strings of any supported class or dtype and
38 0038 1 copies the source string to the destination string, replacing
39 0039 1 any character in the source string that matches a character in
40 0040 1 the match string by the corresponding character in the replace
41 0041 1 string.
42 0042 1
43 0043 1 ENVIRONMENT: User mode, AST level or not or mixed
44 0044 1
45 0045 1 AUTHOR: R. Will, CREATION DATE: 1-Dec-79
46 0046 1
47 0047 1 MODIFIED BY:
48 0048 1
49 0049 1 R. Will, 1-Dec-79: VERSION 01
50 0050 1 1-001 - Original
51 0051 1 1-002 - String speedup, get status from macro. RW 11-Jan-1980
52 0052 1 1-003 - Make trans-table byte, fix signal macro. RW 11-Feb-1980
53 0053 1 1-004 - Enhance to recognize additional classes of descriptors by
54 0054 1 using $STR$GET LEN_ADDR to extract length and address of
55 0055 1 first byte of data from descriptor. Remove string
56 0056 1 code. RKR 29-APR-1981
57 0057 1 1-005 - Expand translate table to 256 bytes so we can accommodate
```

STR\$TRANSLATE
1-008

L 6
16-Sep-1984 01:49:28
14-Sep-1984 12:40:16

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

Page 2
(1)

```
: 58      0058 1 | full 256 character set. RKR 13-MAY-1981
: 59      0059 1 | 1-006 - Speed up code. RKR 7-OCT-1981.
: 60      0060 1 | 1-007 - Use STR$COPY_R_R8 for copy operation. Use $STR$SIGNAL_FATAL
: 61      0061 1 | instead of $STR$CHECK_STATUS. RKR 18-NOV-1981.
: 62      0062 1 | 1-008 - Fix bug introduced by edit 6. Speed up code. Use DECR in creating
: 63      0063 1 | translation table; thus, if a character occurs several times in the
: 64      0064 1 | match string, the first is used for translation. Use STR$COPY_DX_R8
: 65      0065 1 | for the copy operation. PDG 31-Dec-1981
: 66      0066 1 | --
: 67      0067 1 |
: 68      0068 1 | <BLF /PAGE>
```



```

70      0069 1 |
71      0070 1 | SWITCHES:
72      0071 1 |
73      0072 1 |
74      0073 1 | SWITCHES ADDRESSING MODE
75      0074 1 | (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
76      0075 1 |
77      0076 1 |
78      0077 1 | LINKAGES:
79      0078 1 |
80      0079 1 |
81      0080 1 | REQUIRE 'RTLIN:STRLNK';           ! Use require file with string linkage
82      0265 1 |
83      0266 1 |
84      0267 1 | TABLE OF CONTENTS:
85      0268 1 |
86      0269 1 |
87      0270 1 | FORWARD ROUTINE
88      0271 1 | STR$TRANSLATE;                   ! translate routine
89      0272 1 |
90      0273 1 |
91      0274 1 | INCLUDE FILES:
92      0275 1 |
93      0276 1 |
94      0277 1 | REQUIRE 'RTLIN:RTLPSECT';        ! Declare PSECTS code
95      0372 1 |
96      0373 1 | REQUIRE 'RTLIN:STRMACROS';       ! use string macros to code
97      1289 1 |
98      1290 1 | LIBRARY 'RTLSTARLE';            ! STARLET library for macros and symbol
99      1291 1 |
100     1292 1 |
101     1293 1 | MACROS: NONE
102     1294 1 |
103     1295 1 |
104     1296 1 |
105     1297 1 | EQUATED SYMBOLS: NONE
106     1298 1 |
107     1299 1 |
108     1300 1 |
109     1301 1 | PSECT DECLARATIONS
110     1302 1 |
111     1303 1 |
112     1304 1 | DECLARE_PSECTS (STR);
113     1305 1 |
114     1306 1 |
115     1307 1 | OWN STORAGE: NONE
116     1308 1 |
117     1309 1 |
118     1310 1 |
119     1311 1 | EXTERNAL REFERENCES:
120     1312 1 |
121     1313 1 |
122     1314 1 | EXTERNAL ROUTINE
123     1315 1 | LIB$STOP,                        ! Signal errors
124     1316 1 | STR$COPY_DX_R8 : STR$JSB_COPY_DX ; ! Routine to do the copy
125     1317 1 |
126     1318 1 | EXTERNAL LITERAL
```

STR\$TRANSLATE
1-008

; 127

1319 1

STR\$_NORMAL;

N 6
16-Sep-1984 01:49:28
14-Sep-1984 12:40:16
! successful completion

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

Page 4
(2)

STR
1-(

.....


```
129 1320 1 GLOBAL ROUTINE STR$TRANSLATE ( ! translate
130 1321 1
131 1322 1     DEST_DESC, ! Pointer to destination descriptor
132 1323 1     SRC_DESC, ! Pointer to source descriptor
133 1324 1     TRANSLATE_DESC, ! Pointer to translation string desc
134 1325 1     MATCH_DESC ! pointer to match string desc
135 1326 1
136 1327 1     ) : =
137 1328 1
138 1329 1 ++
139 1330 1 FUNCTIONAL DESCRIPTION:
140 1331 1     This routine copies the input string to the output string and
141 1332 1     translates all characters in the source string which are also in
142 1333 1     the match string to the corresponding character in the
143 1334 1     translate string. If the translate string is shorter than the
144 1335 1     match string, the replace string is expanded to the length of
145 1336 1     the match string with blanks.
146 1337 1
147 1338 1 FORMAL PARAMETERS:
148 1339 1
149 1340 1     DEST_DESC.wt.dx pointer to destination string descriptor
150 1341 1     SRC_DESC.rt.dx pointer to source string descriptor
151 1342 1     TRANSLATE_DESC.rt.dx pointer to translate string descriptor
152 1343 1     MATCH_DESC.rt.dx pointer to match string descriptor
153 1344 1
154 1345 1 IMPLICIT INPUTS:
155 1346 1
156 1347 1     NONE
157 1348 1
158 1349 1 IMPLICIT OUTPUTS:
159 1350 1
160 1351 1     NONE
161 1352 1
162 1353 1 COMPLETION CODES:
163 1354 1
164 1355 1     SSS$NORMAL Success
165 1356 1     STR$_TRU Truncation occurred. Warning.
166 1357 1
167 1358 1 SIDE EFFECTS:
168 1359 1
169 1360 1     May signal:
170 1361 1     STR$_FATINTERR Fatal internal error
171 1362 1     STR$_ILLSTRCLA Illegal (or unsupported) string class
172 1363 1     STR$_INSVIRMEM Insufficient virtual memory for
173 1364 1     reallocation of dynamic string
174 1365 1
175 1366 1 --
176 1367 1
177 1368 2 BEGIN
178 1369 2
179 1370 2 LOCAL
180 1371 2     MATCH_LEN, ! length of match string
181 1372 2     MATCH_ADDR, ! addr of 1st byte of
182 1373 2     match string
183 1374 2     TRANS_LEN, ! length of translate
184 1375 2     string
185 1376 2     TRANS_ADDR, ! addr of 1st byte of
```



```
186 1377 2      DEST_LEN,      ! translate string
187 1378 2      ! length of destination
188 1379 2      ! string
189 1380 2      DEST_ADDR,    ! addr of 1st byte of
190 1381 2      ! destination string
191 1382 2      MIN_LEN,
192 1383 2      RETURN_STATUS, ! status from STR$COPY
193 1384 2      TRANSLATE_TABLE : VECTOR [256, BYTE]; ! table to do translate
194 1385 2
195 1386 2      MAP
196 1387 2      DEST_DESC      : REF $STR$DESCRIPTOR,
197 1388 2      SRC_DESC        : REF $STR$DESCRIPTOR,
198 1389 2      TRANSLATE_DESC : REF $STR$DESCRIPTOR,
199 1390 2      MATCH_DESC     : REF $STR$DESCRIPTOR;
200 1391 2
201 1392 2      !+
202 1393 2      ! compute length and address of 1st data byte of strings involved
203 1394 2      ! and signal if any fatal errors result
204 1395 2      !-
205 1396 2      $STR$GET_LEN_ADDR ( MATCH_DESC, MATCH_LEN, MATCH_ADDR ) ;
206 1397 2      $STR$GET_LEN_ADDR ( TRANSLATE_DESC, TRANS_LEN, TRANS_ADDR ) ;
207 1398 2
208 1399 2      !+
209 1400 2      ! Copy source to the destination string and signal fatal errors.
210 1401 2      !-
211 1402 2
212 1403 2      RETURN_STATUS = STR$COPY_DX_R8 ( .DEST_DESC, .SRC_DESC );
213 1404 2
214 1405 2      $STR$SIGNAL_FATAL (RETURN_STATUS) ;
215 1406 2
216 1407 2      !+
217 1408 2      ! Get length and addr of 1st byte of destination after copy operation.
218 1409 2      !-
219 1410 2      $STR$GET_LEN_ADDR ( DEST_DESC, DEST_LEN, DEST_ADDR ) ;
220 1411 2      !+
221 1412 2      ! The copy is finished, now create the translate table
222 1413 2      !-
223 1414 2
224 1415 2      BEGIN      ! This block does the equivalent of:
225 1416 2      !      INCR I FROM 0 TO 255 DO TRANSLATE_TABLE[I]=.I ;
226 1417 2      !      but does it a longword at a time
227 1418 2      LOCAL
228 1419 2      X: REF VECTOR;
229 1420 2      X = TRANSLATE_TABLE;
230 1421 2      X[0] = %X'03020100';
231 1422 2      DECR I FROM 256/XUPVAL-2 TO 0 DO      ! initilize to translate to self
232 1423 4      BEGIN
233 1424 4      X[I] = .X[0] + %X'04040404';
234 1425 4      X = .X + %UPVAL;
235 1426 3      END;
236 1427 2      END;
237 1428 2
238 1429 2      MIN_LEN = MINU(.MATCH_LEN, .TRANS_LEN);
239 1430 2
240 1431 2      INCR I FROM .MIN_LEN TO .MATCH_LEN-1 DO
241 1432 2      TRANSLATE_TABLE [CH$RCHAR (CH$PLUS (.MATCH_ADDR, .I))] =
242 1433 2      STR$K_FILL_CHAR;
```



```
243 1434 2
244 1435 2
245 1436 2
246 1437 2
247 1438 2
248 1439 2
249 1440 2
250 1441 2
251 1442 2
252 1443 2
253 1444 2
254 1445 2
255 1446 2
256 1447 2
257 1448 2
258 1449 2
259 1450 2
260 1451 2
261 1452 2
262 1453 1

      DECR I FROM .MIN_LEN-1 TO 0 DO      ! fill in users translate
      TRANSLATE_TABLE [CH$RCHAR (CH$PLUS (.MATCH_ADDR, .I))] =
      CH$RCHAR (CH$PLUS (.TRANS_ADDR, .I));

      +
      - Now translate the destination string in place.

      CH$TRANSLATE (
      TRANSLATE_TABLE,                    ! ASCII translation table
      .DEST_LEN,                          ! from destination
      .DEST_ADDR,                         !
      STR$K_FILL_CHAR,                   ! never used since same string
      .DEST_LEN,                          ! to destination
      .DEST_ADDR);

      RETURN .RETURN_STATUS;
      END;

      !End of STR$TRANSLATE
```

```

      .TITLE STR$TRANSLATE
      .IDENT \1-008\

      .EXTRN LIB$STOP, STR$COPY_DX_R8
      .EXTRN STR$NORMAL, STR$ANALYZE_SDESC_R1

      .PSECT _STR$CODE, NOWRT, SHR, PIC, 2

      .ENTRY STR$TRANSLATE, Save R2,R3,R4,R5,R6,R7,R8,-
      R9,R10,R11
      MOVAB -264(SP), SP
      MOVL MATCH_DESC, R0
      CMPB 3(R0), #2
      BGTRU 1$
      MOVZWL (R0), MATCH_LEN
      MOVL 4(R0), MATCH_ADDR
      BRB 2$
      JSB STR$ANALYZE_SDESC_R1
      MOVL R0, 4(SP)
      MOVL R1, R10
      MOVL TRANSLATE_DESC, R0
      CMPB 3(R0), #2
      BGTRU 3$
      MOVZWL (R0), TRANS_LEN
      MOVL 4(R0), TRANS_ADDR
      BRB 4$
      JSB STR$ANALYZE_SDESC_R1
      MOVL R0, (SP)
      MOVL R1, R11
      MOVL DEST_DESC, R9
      MOVL SRC_DESC, R1
      MOVL R9, R0
      JSB STR$COPY_DX_R8
      MOVL R0, RETURN_STATUS
      BLBS RETURN_STATUS, 5$

      OFFC 00000
      5E FEF8 CE 9E 00002
      50 10 AC D0 00007
      02 03 A0 91 0000B
      04 AE 04 60 1A 0000F
      5A 04 A0 3C 00011
      04 AE 00000000G 0D 11 00019
      5A 04 00 16 0001B 1$:
      50 D0 00021
      5A 51 D0 00025
      50 0C AC D0 00028 2$:
      02 03 A0 91 0002C
      09 1A 00030
      6E 60 3C 00032
      5B 04 A0 D0 00035
      0C 11 00039
      04 AE 00000000G 00 16 0003B 3$:
      5B 50 D0 00041
      59 04 51 D0 00044
      51 08 AC D0 00047 4$:
      50 59 D0 0004F
      04 AE 00000000G 00 16 00052
      56 50 D0 00058
      10 56 E8 0005B
```

```

      .TITLE STR$TRANSLATE
      .IDENT \1-008\

      .EXTRN LIB$STOP, STR$COPY_DX_R8
      .EXTRN STR$NORMAL, STR$ANALYZE_SDESC_R1

      .PSECT _STR$CODE, NOWRT, SHR, PIC, 2

      .ENTRY STR$TRANSLATE, Save R2,R3,R4,R5,R6,R7,R8,-
      R9,R10,R11
      MOVAB -264(SP), SP
      MOVL MATCH_DESC, R0
      CMPB 3(R0), #2
      BGTRU 1$
      MOVZWL (R0), MATCH_LEN
      MOVL 4(R0), MATCH_ADDR
      BRB 2$
      JSB STR$ANALYZE_SDESC_R1
      MOVL R0, 4(SP)
      MOVL R1, R10
      MOVL TRANSLATE_DESC, R0
      CMPB 3(R0), #2
      BGTRU 3$
      MOVZWL (R0), TRANS_LEN
      MOVL 4(R0), TRANS_ADDR
      BRB 4$
      JSB STR$ANALYZE_SDESC_R1
      MOVL R0, (SP)
      MOVL R1, R11
      MOVL DEST_DESC, R9
      MOVL SRC_DESC, R1
      MOVL R9, R0
      JSB STR$COPY_DX_R8
      MOVL R0, RETURN_STATUS
      BLBS RETURN_STATUS, 5$

      1320
      1396
      1397
      1403
      1405
```

04	56	03	00	ED	0005E	CMPZV	#0, #3, RETURN_STATUS, #4	:	
			09	12	00063	BNEQ	5\$	:	
	00000000G	00	56	DD	00065	PUSHL	RETURN STATUS	:	
		02	01	FB	00067	CALLS	#1, LIB\$STOP	:	
		03	A9	91	0006E	5\$: CMPB	3(R9), #2	:	1410
			09	1A	00072	BGTRU	6\$	:	
		54	69	3C	00074	MOVZWL	(R9), DEST_LEN	:	
		51	A9	D0	00077	MOVL	4(R9), DEST_ADDR	:	
			0C	11	0007B	BRB	7\$	:	
		50	59	D0	0007D	6\$: MOVL	R9, R0	:	
			00	16	0008C	JSB	STR\$ANALYZE_SDESC_R1	:	
		54	50	D0	00086	MOVL	R0, R4	:	
		50	08	AE	9E	7\$: MOVAB	TRANSLATE_TABLE, X	:	1420
		60	03020100	8F	D0	0008D	MOVL	#50462976, (X)	1421
		52		3E	D0	00094	MOVL	#62, I	1422
	60	80	04040404	8F	C1	00097	8\$: ADDL3	#67372036, (X)+, (X)	1424
		F5		52	F4	0009F	SOBGEQ	I, 8\$	1422
		50		AE	D0	000A2	MOVL	MATCH_LEN, R0	1429
		6E		50	D1	000A6	CMPL	R0, TRANS_LEN	
				03	1B	000A9	BLEQU	9\$	
		50		6E	D0	000AB	MOVL	TRANS_LEN, R0	
		52		50	D0	000AE	9\$: MOVL	R0, MIN_LEN	
		50	FF	A2	9E	000B1	MOVAB	-1(R2), I	1432
				09	11	000B5	BRB	11\$	
		53		604A	9A	000B7	10\$: MOVZBL	(I)[MATCH_ADDR], R3	
		08 AE43		20	90	000BB	MOVB	#32, TRANSLATE_TABLE[R3]	
	F2	50	04	AE	F2	000C0	11\$: AOBLS	MATCH_LEN, I, TOS	
		50		52	D0	000C5	MOVL	MIN_LEN, I	1438
				0A	11	000C8	BRB	13\$	
		52		604A	9A	000CA	12\$: MOVZBL	(I)[MATCH_ADDR], R2	1437
		08 AE42		604B	90	000CE	MOVB	(I)[TRANS_ADDR], TRANSLATE_TABLE[R2]	1438
		F3		50	F4	000D4	13\$: SOBGEQ	I, 12\$	1437
08 AE	20	61		54	2E	000D7	MOVTC	DEST_LEN, (DEST_ADDR), #32, -	1450
		61		54		000DD		TRANSLATE_TABLE, DEST_LEN, (DEST_ADDR)	
		50		56	D0	000DF	MOVL	RETURN_STATUS, R0	1452
				04	000E2	RET			1453

; Routine Size: 227 bytes, Routine Base: _STR\$CODE + 0000

: 263 1454 1
: 264 1455 1 END
: 265 1456 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
_STR\$CODE	227	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	6	0	581	00:00.7

COMMAND QUALIFIERS

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

; Size: 227 code + 0 data bytes
; Run Time: 00:06.5
; Elapsed Time: 00:27.5
; Lines/CPU Min: 13378
; Lexemes/CPU-Min: 34980
; Memory Used: 110 pages
; Compilation Complete

0215 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

