

[illegible]


```
1 0001 0 MODULE STRSTRIM ( ! Delete trailing blanks and spaces
2 0002 0
3 0003 0 IDENT = '1-007' ! File: STRTRIM.B32 Edit: RKR1007
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 deletes trailing spaces and tabs.
39 0039 1
40 0040 1 ENVIRONMENT: User mode, AST level or not or mixed
41 0041 1
42 0042 1 AUTHOR: R. Will, CREATION DATE: 19-Feb-79
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 R. Will, 19-Feb-79: VERSION 01
47 0047 1 1-001 - Separate from BAS$EDIT, and write in B32 instead of MAR.
48 0048 1 RW 5-Nov-79
49 0049 1 1-002 - Make optional argument word. RW 8-Nov-79
50 0050 1 1-003 - String speedup, status from macros. RW 11-Jan-1980
51 0051 1 1-004 - Compute optional output length argument after the copy to the
52 0052 1 result so that the correct output string length is used.
53 0053 1 SBL 26-June-1981
54 0054 1 1-005 - Enhance to recognize additional classes of descriptors by
55 0055 1 using $STR$GET_LEN_ADDR to extract the length and address
56 0056 1 of the first byte of data from a descriptor. Remove string
57 0057 1 interlocking code. RKR 24-AUG-1981
```

STRTRIM
1-007

1 7
16-Sep-1984 01:50:13
14-Sep-1984 12:40:16

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

Page 2
(1)

```
: 58      0058 1 ! 1-006 - Speed up code. RKR 7-OCT-1981.
: 59      0059 1 ! 1-007 - Use STR$COPY_R R8 for copying operation. Use
: 60      0060 1 ! $STR$SIGNAL FATAL instead of $STR$CHECK_STATUS.
: 61      0061 1 ! RKR 18-NOV-1981.
: 62      0062 1 ! --
: 63      0063 1 !
: 64      0064 1 ! <BLF/PAGE>
```



```

66      0065 1 |
67      0066 1 | SWITCHES:
68      0067 1 |
69      0068 1 |
70      0069 1 SWITCHES ADDRESSING_MODE
71      0070 1          (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
72      0071 1 |
73      0072 1 |
74      0073 1 | LINKAGES:
75      0074 1 |
76      0075 1 |
77      0076 1 REQUIRE 'RTLIN:STRLNK';           ! Use require file with string linkage
78      0261 1 |
79      0262 1 |
80      0263 1 | TABLE OF CONTENTS:
81      0264 1 |
82      0265 1 |
83      0266 1 FORWARD ROUTINE
84      0267 1     STR$TRIM;                     ! delete trailing spaces and tabs
85      0268 1 |
86      0269 1 |
87      0270 1 | INCLUDE FILES:
88      0271 1 |
89      0272 1 |
90      0273 1 REQUIRE 'RTLIN:RTLPSECT';         ! Declare PSECTS code
91      0368 1 |
92      0369 1 REQUIRE 'RTLIN:STRMACROS';        ! use string macros to code
93      1285 1 |
94      1286 1 LIBRARY 'RTLSTARLE';             ! STARLET library for macros and symbol
95      1287 1 |
96      1288 1 |
97      1289 1 | MACROS: NONE
98      1290 1 |
99      1291 1 |
100     1292 1 |
101     1293 1 | EQUATED SYMBOLS:
102     1294 1 |
103     1295 1 |
104     1296 1 LITERAL
105     1297 1     SPACE = ' ', ' ', ;          ! space
106     1298 1     TAB = ' ', ' ', ;            ! tab
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

```

STRSTRIM
1-007

K 7
16-Sep-1984 01:50:13 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:16 [LIBRTL.SRC]STRTRIM.B32;1

Page 4
(2)

```
: 123      1315 1    LIB$STOP,  
: 124      1316 1    STR$COPY_R_R8 : STR$JSB_COPY_R ;  
: 125      1317 1  
: 126      1318 1  EXTERNAL LITERAL  
: 127      1319 1    STR$_NORMAL;  
  
! signal errors  
! Routine to do the copy  
  
! successful completion
```



```
129 1320 1 GLOBAL ROUTINE STRSTRIM (      ! delete trailing spaces and tabs
130 1321 1
131 1322 1     DEST_DESC,      ! Pointer to destination descriptor
132 1323 1     SRC_DESC,      ! Pointer to source descriptor
133 1324 1     DEST_LEN      ! length of destination
134 1325 1
135 1326 1     ) : =
136 1327 1
137 1328 1 ++
138 1329 1 FUNCTIONAL DESCRIPTION:
139 1330 1
140 1331 1     This routine copies the input string to the output string
141 1332 1     without copying the trailing spaces and tabs.  If the optional
142 1333 1     third argument is present, the length of the copy is returned.
143 1334 1
144 1335 1 FORMAL PARAMETERS:
145 1336 1
146 1337 1     DEST_DESC.wt.dx    pointer to destination string descriptor
147 1338 1     SRC_DESC.rt.dx     pointer to source string descriptor
148 1339 1     DEST_LEN.wvu.r     length of trimmed source string
149 1340 1
150 1341 1 IMPLICIT INPUTS:
151 1342 1
152 1343 1     NONE
153 1344 1
154 1345 1 IMPLICIT OUTPUTS:
155 1346 1
156 1347 1     NONE
157 1348 1
158 1349 1 COMPLETION CODES:
159 1350 1
160 1351 1     SS$ NORMAL        Success
161 1352 1     STR$_TRU          Truncation occurred.  Warning.
162 1353 1
163 1354 1 SIDE EFFECTS:
164 1355 1
165 1356 1     May signal:
166 1357 1         STR$_FATINTERR  Fatal internal error
167 1358 1         STR$_ILLSTRCLA  Illegal (or unsupported) string class
168 1359 1         STR$_INSVIRMEM  Insufficient virtual memory for
169 1360 1                        reallocation of dynamic string
170 1361 1
171 1362 1 --
172 1363 1
173 1364 2 BEGIN
174 1365 2
175 1366 2 BUILTIN
176 1367 2     NULLPARAMETER;      ! check for optional arg
177 1368 2
178 1369 2 LOCAL
179 1370 2     IN_LEN,              ! length of input string
180 1371 2     IN_ADDR,          ! addr of 1st byte of
181 1372 2                     ! input string
182 1373 2     OUT_LEN,          ! length of output
183 1374 2                     ! string
184 1375 2     OUT_ADDR,         ! addr of 1st data byte
185 1376 2                     ! of output string
```

```
186 1377 2      RETURN_STATUS,      ! status from
187 1378 2      COPY_LENGTH;      ! STR$COPY_R_R8
188 1379 2      ! length to be copied
189 1380 2
190 1381 2      MAP
191 1382 2      SRC_DESC : REF $STR$DESCRIPTOR,
192 1383 2      DEST_DESC : REF $STR$DESCRIPTOR,
193 1384 2      DEST_LEN : REF VECTOR [1, WORD, UNSIGNED]; ! make parameter a
194 1385 2      ! word
195 1386 2
196 1387 2      !+
197 1388 2      ! compute length and address of 1st data byte for input string and
198 1389 2      ! signal fatal errors encountered.
199 1390 2      !-
200 1391 2      $STR$GET_LEN_ADDR ( SRC_DESC, IN_LEN, IN_ADDR ) ;
201 1392 2
202 1393 2      !+
203 1394 2      ! Start at the end of the string and search backwards to find the first
204 1395 2      ! character which is not a tab or a space.
205 1396 2      !-
206 1397 2
207 1398 2      COPY_LENGTH = .IN_LEN;      ! initialize the length
208 1399 2
209 1400 2      WHILE .COPY_LENGTH NEQU 0
210 1401 2      DO
211 1402 2      BEGIN
212 1403 2      IF
213 1404 2      ((CH$RCHAR (CH$PLUS (.IN_ADDR, .COPY_LENGTH - 1))
214 1405 2      EQL SPACE)
215 1406 2      OR
216 1407 2      (CH$RCHAR (CH$PLUS (.IN_ADDR, .COPY_LENGTH - 1))
217 1408 2      EQL TAB))
218 1409 2      THEN
219 1410 2      COPY_LENGTH = .COPY_LENGTH - 1
220 1411 2      ELSE
221 1412 2      EXITLOOP;
222 1413 2      END;
223 1414 2
224 1415 2      !+
225 1416 2      ! Copy the trimmed string
226 1417 2      !-
227 1418 2
228 1419 2      RETURN_STATUS = STR$COPY_R_R8 ( .DEST_DESC,
229 1420 2      .COPY_LENGTH,
230 1421 2      .IN_ADDR ) ;
231 1422 2
232 1423 2      $STR$SIGNAL_FATAL (RETURN_STATUS);      ! signal fatal errors
233 1424 2
234 1425 2      !+
235 1426 2      ! If caller asked for length to be returned, return it. In this case,
236 1427 2      ! we have to use $STR$GET_LEN_ADDR to extract its length.
237 1428 2      !-
238 1429 2
239 1430 2      IF NOT NULLPARAMETER (3)
240 1431 2      THEN
241 1432 2      BEGIN
242 1433 2      $STR$GET_LEN_ADDR ( DEST_DESC, OUT_LEN, OUT_ADDR ) ;
```


STR\$TRIM
1-007

N 7
16-Sep-1984 01:50:13
14-Sep-1984 12:40:16

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]STR\$TRIM.B32;1

Page 7
(3)

: 243 1434 3
: 244 1435 2
: 245 1436 2
: 246 1437 2
: 247 1438 1

DEST_LEN [0] = MIN (.OUT_LEN, .COPY_LENGTH);
END ;
RETURN .RETURN_STATUS;
END;

!End of STR\$TRIM

.TITLE STR\$TRIM
.IDENT \1-007\

.EXTRN LIB\$STOP, STR\$COPY_R_R8
.EXTRN STR\$_NORMAL, STR\$ANALYZE_SDESC_R1

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

.ENTRY STR\$TRIM, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; 1320
R11

MOVAB STR\$ANALYZE_SDESC_R1, R11
MOVL SRC_DESC, R0 1391

CMPB 3(R0), #2

BGTRU 1\$

MOVZWL (R0), IN_LEN

MOVL 4(R0), IN_ADDR

BRB 2\$

JSB STR\$ANALYZE_SDESC_R1

MOVL R0, R3

MOVL R1, R2

MOVL IN_LEN, COPY_LENGTH 1398

BEQL 5\$ 1400

CMPB -1(COPY_LENGTH)[IN_ADDR], #32 1405

BEQL 4\$

CMPB -1(COPY_LENGTH)[IN_ADDR], #9 1408

BNEQ 5\$

DECL COPY_LENGTH 1410

BRB 3\$

MOVL DEST_DESC, R9 1419

MOVQ R9, R0

JSB STR\$COPY_R_R8

MOVL R0, RETURN_STATUS

BLBS RETURN_STATUS, 6\$ 1423

CMPZV #0, #3, RETURN_STATUS, #4

BNEQ 6\$

PUSHL RETURN_STATUS

CALLS #1, LIB\$STOP

CMPB (AP), #3 1430

BLSSU 10\$

TSTL 12(AP)

BEQL 10\$

CMPB 3(R9), #2 1433

BGTRU 7\$

MOVZWL (R9), OUT_LEN

MOVL 4(R9), OUT_ADDR

BRB 8\$

MOVL R9, R0

JSB STR\$ANALYZE_SDESC_R1

CMPL R0, COPY_LENGTH 1434

BLEQ 9\$

04

52

00000000G

0C

02

50

51

50

5A

OFFC 00000

5B 00000000G 00 9E 00002

50 08 AC D0 00009

02 03 A0 91 0000D

09 1A 00011

53 60 3C 00013

52 04 A0 D0 00016

08 11 0001A

6B 16 0001C 1\$:

53 50 D0 0001E

52 51 D0 00021

5A 53 D0 00024 2\$:

12 13 00027 3\$:

20 FF AA42 91 00029

07 13 0002E

09 FF AA42 91 00030

04 12 00035

5A D7 00037 4\$:

EC 11 00039

59 04 AC D0 0003B 5\$:

50 59 7D 0003F

00000000G 00 16 00042

52 50 D0 00048

10 52 E8 0004B

03 00 ED 0004E

09 12 00053

52 DD 00055

00 01 FB 00057

03 6C 91 0005E 6\$:

25 1F 00061

AC D5 00063

20 13 00066

02 03 A9 91 00068

09 1A 0006C

50 69 3C 0006E

51 04 A9 D0 00071

05 11 00075

50 59 D0 00077 7\$:

6B 16 0007A

5A 50 D1 0007C 8\$:

03 15 0007F

STRSTRIM
1-007

B 8
16-Sep-1984 01:50:13
14-Sep-1984 12:40:16

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

Page 8
(3)

OC	50	5A	D0	00081	MOVL	COPY_LENGTH, R0
	BC	50	B0	00084	MOVW	R0, DEST_LEN
	50	52	D0	00088	MOVL	RETURN_STATUS, R0
		04	0008B		RET	

1437
1438

; Routine Size: 140 bytes, Routine Base: _STR\$CODE + 0000

: 248	1439	1	
: 249	1440	1	END
: 250	1441	1	
: 251	1442	0	ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
_STR\$CODE	140	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	00:00.8

COMMAND QUALIFIERS

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

; Size: 140 code + 0 data bytes
; Run Time: 00:05.8
; Elapsed Time: 00:20.0
; Lines/CPU Min: 14815
; Lexemes/CPU-Min: 35547
; Memory Used: 96 pages
; Compilation Complete

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

