


```

1 0001 0 MODULE STR$COMPARE_EQL (      ! Compare 2 strings-contents and length
2 0002 0
3 0003 0      IDENT = '1-005' ! File: STR$COMEQL.B32 Edit RKR1005
4 0004 0
5 0005 0      ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 |
9 0009 1 |*****|
10 0010 1 |*|
11 0011 1 |*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY |*|
12 0012 1 |*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. |*|
13 0013 1 |*  ALL RIGHTS RESERVED. |*|
14 0014 1 |*|
15 0015 1 |*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED |*|
16 0016 1 |*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE |*|
17 0017 1 |*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER |*|
18 0018 1 |*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY |*|
19 0019 1 |*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY |*|
20 0020 1 |*  TRANSFERRED. |*|
21 0021 1 |*|
22 0022 1 |*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE |*|
23 0023 1 |*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT |*|
24 0024 1 |*  CORPORATION. |*|
25 0025 1 |*|
26 0026 1 |*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS |*|
27 0027 1 |*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. |*|
28 0028 1 |*|
29 0029 1 |*|
30 0030 1 |*****|
31 0031 1 |
32 0032 1 |
33 0033 1 |**|
34 0034 1 | FACILITY: String support library |
35 0035 1 |
36 0036 1 | ABSTRACT: This module takes 2 input strings of any supported class |
37 0037 1 | and dtype and returns a 0 if string1 is the same as string2 in both |
38 0038 1 | length and contents and a 1 otherwise |
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: 26-Mar-79 |
43 0043 1 |
44 0044 1 | MODIFIED BY: |
45 0045 1 |
46 0046 1 | R. Will, 26-Mar-79 : VERSION 01 |
47 0047 1 | 1-001 - Original |
48 0048 1 | 1-002 - String cleanup, change name to STR$COMEQL. RW 5-Nov-79 |
49 0049 1 | 1-003 - Make routine sensitive to additional classes of string |
50 0050 1 | descriptors by using $STR$GET_LEN_ADDR to extract length and |
51 0051 1 | starting addresses from descriptors. Remove string |
52 0052 1 | interlocking code. RKR 13-APR-81 |
53 0053 1 | 1-004 - Speed up code. RKR 13-OCT-1981. |
54 0054 1 | 1-005 - Use SWITCHES ZIP to avoid some of the cross-jumping code |
55 0055 1 | that would normally get generated this module. |
56 0056 1 | RKR 18-NOV-1981. |
57 0057 1 | --

```

; R

STR\$COMPARE_EQL
1-005

1
16-Sep-1984 01:32:27
14-Sep-1984 12:40:01

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

Page 2
(1)

; 58 0058 1 !<BLF/PAGE>

STR
1-0

:

.....

.....

.....

.....
S
.....

```
60 0059 1 |  
61 0060 1 | SWITCHES:  
62 0061 1 |  
63 0062 1 |  
64 0063 1 | SWITCHES ADDRESSING MODE  
65 0064 1 | (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);  
66 0065 1 |  
67 0066 1 | SWITCHES ZIP ;  
68 0067 1 |  
69 0068 1 | LINKAGES  
70 0069 1 |  
71 0070 1 | REQUIRE 'RTLIN:STRLNK'; ! Linkage to LIB$ANALYZE_DESC_R3  
72 0255 1 |  
73 0256 1 |  
74 0257 1 | TABLE OF CONTENTS:  
75 0258 1 |  
76 0259 1 |  
77 0260 1 | FORWARD ROUTINE  
78 0261 1 | STR$COMPARE_EQL; ! compare 2 strings, same length and contents  
79 0262 1 |  
80 0263 1 |  
81 0264 1 | INCLUDE FILES:  
82 0265 1 |  
83 0266 1 |  
84 0267 1 | REQUIRE 'RTLIN:RTLPSECT'; ! Declare PSECTS code  
85 0362 1 | REQUIRE 'RTLIN:STRMACROS'; ! use string macros to write code  
86 1278 1 | LIBRARY 'RTLSTARLE'; ! STARLET library for macros and symbols  
87 1279 1 |  
88 1280 1 |  
89 1281 1 | MACROS: NONE  
90 1282 1 |  
91 1283 1 |  
92 1284 1 |  
93 1285 1 | EQUATED SYMBOLS: NONE  
94 1286 1 |  
95 1287 1 |  
96 1288 1 |  
97 1289 1 | PSECT DECLARATIONS  
98 1290 1 |  
99 1291 1 |  
100 1292 1 | DECLARE_PSECTS (STR);  
101 1293 1 |  
102 1294 1 |  
103 1295 1 | OWN STORAGE: NONE  
104 1296 1 |  
105 1297 1 |  
106 1298 1 |  
107 1299 1 | EXTERNAL REFERENCES  
108 1300 1 |  
109 1301 1 | NONE
```

```
111 1302 1 GLOBAL ROUTINE STR$COMPARE_EQL ( ! compare strings-same len and contents
112 1303 1
113 1304 1     STRING1, ! pointer to 1st string descriptor
114 1305 1     STRING2 ! pointer to 2nd string descriptor
115 1306 1
116 1307 1     ) : =
117 1308 1
118 1309 1 ++
119 1310 1 FUNCTIONAL DESCRIPTION:
120 1311 1
121 1312 1     This routine takes two source strings of any supported
122 1313 1     dtype and class, compares their length and contents and returns
123 1314 1     a 0 if both are the same (length and contents)
124 1315 1     and a 1 otherwise.
125 1316 1
126 1317 1 FORMAL PARAMETERS:
127 1318 1
128 1319 1     STRING1.rt.dx     pointer to 1st string descriptor
129 1320 1     STRING2.rt.dx     pointer to 2nd string descriptor
130 1321 1
131 1322 1 IMPLICIT INPUTS:
132 1323 1
133 1324 1     NONE
134 1325 1
135 1326 1 IMPLICIT OUTPUTS:
136 1327 1
137 1328 1     NONE
138 1329 1
139 1330 1 ROUTINE VALUE:
140 1331 1
141 1332 1     MATCH.wl.v       0 if same length and contents, otherwise 1
142 1333 1
143 1334 1 SIDE EFFECTS:
144 1335 1
145 1336 1     May signal STR$_ILLSTRCLA on bad string class
146 1337 1
147 1338 1 --
148 1339 1
149 1340 2 BEGIN
150 1341 2
151 1342 2 LOCAL
152 1343 2     MATCH;           ! Local to return match or nomatch
153 1344 2
154 1345 2     MAP STRING1 : REF $STR$DESCRIPTOR;
155 1346 2     MAP STRING2 : REF $STR$DESCRIPTOR;
156 1347 2
157 1348 2 ++
158 1349 2 Find out if this is a static or dynamic string to optimize the code
159 1350 2 path
160 1351 2 --
161 1352 2 IF (.STRING1 [DSC$B_CLASS] LEQU DSC$K_CLASS_D) AND
162 1353 2     (.STRING2 [DSC$B_CLASS] LEQU DSC$K_CLASS_D)
163 1354 2 THEN
164 1355 2     BEGIN
165 1356 2     ++
166 1357 2     ! If lengths are equal and contents are same match = 0,
167 1358 2     ! else match = 1
```

```

168 1359 3  !-
169 1360 3
170 1361 4  IF (.STRING1 [DSC$W_LENGTH] EQLU .STRING2 [DSC$W_LENGTH] )
171 1362 3  THEN ! lengths equal
172 1363 4  BEGIN
173 1364 4  +
174 1365 4  | Use same length field to force generation of a CMPC3. This
175 1366 4  | is ok since lengths are the same.
176 1367 4  -
177 1368 5  IF (CH$EQL
178 1369 5  (.STRING1 [DSC$W_LENGTH], .STRING1 [DSC$A_POINTER],
179 1370 5  .STRING1 [DSC$W_LENGTH], .STRING2 [DSC$A_POINTER] ) )
180 1371 4  THEN MATCH = 0 ! both length and contents match
181 1372 4  ELSE MATCH = 1 ! lengths match, contents don't
182 1373 4  END
183 1374 3  ELSE MATCH = 1; ! lengths don't match
184 1375 3
185 1376 3  END
186 1377 3
187 1378 2  ELSE
188 1379 2  BEGIN
189 1380 2  LOCAL
190 1381 2  S1_LENGTH, ! length of first string
191 1382 2  S1_ADDR, ! address of 1st data byte of first string
192 1383 2  S2_LENGTH, ! length of second string
193 1384 2  S2_ADDR; ! address of 1st data byte of second string
194 1385 2
195 1386 2  +
196 1387 2  | Get the associated lengths and addresses of the strings
197 1388 2  |
198 1389 2  |
199 1390 2  $STR$GET_LEN_ADDR (STRING1, S1_LENGTH, S1_ADDR) ;
200 1391 2  $STR$GET_LEN_ADDR (STRING2, S2_LENGTH, S2_ADDR) ;
201 1392 2  |
202 1393 2  +
203 1394 2  | If lengths are equal and contents are same match = 0,
204 1395 2  | else match = 1
205 1396 2  |
206 1397 2  |
207 1398 4  IF (.S1_LENGTH EQLU .S2_LENGTH ) ! lengths = ?
208 1399 3  THEN
209 1400 4  BEGIN
210 1401 5  IF (CH$EQL (.S1_LENGTH, .S1_ADDR, ! contents = ?
211 1402 5  .S1_LENGTH, .S2_ADDR) ) ! use same length field
212 1403 5  | to force generation of
213 1404 5  | CMPC3 compare. Its ok
214 1405 5  | since we know
215 1406 5  | S1_LENGTH = S2_LENGTH
216 1407 4  THEN MATCH = 0 ! both length and contents match
217 1408 4  ELSE MATCH = 1 ! lengths match, contents don't
218 1409 4  END
219 1410 3  ELSE MATCH = 1; ! lengths don't match
220 1411 3
221 1412 2  END;
222 1413 2
223 1414 2  RETURN .MATCH;
224 1415 1  END; ! End of STR$COMPARE_EQL

```

					.TITLE	STR\$COMPARE_EQL		
					.IDENT	\1-005\		
					.EXTRN	STR\$ANALYZE_SDESC_R1		
					.PSECT	_STR\$CODE,NOWRT, SHR, PIC,2		
					.ENTRY	STR\$COMPARE_EQL, Save R2,R3,R4,R5		1302
54	08	AC	D0	00002	MOVL	STRING2, R4		1361
50	C4	AC	D0	00006	MOVL	STRING1, R0		1352
		52	D4	0000A	CLRL	R2		
02	03	A0	91	0000C	CMPB	3(R0), #2		
		19	1A	00010	BGTRU	1\$		
		52	D6	00012	INCL	R2		
51	08	AC	D0	00014	MOVL	STRING2, R1		1353
02	03	A1	91	00018	CMPB	3(R1), #2		
		0D	1A	0001C	BGTRU	1\$		
64		60	B1	0001E	CMPW	(R0), (R4)		1361
		44	12	00021	BNEQ	7\$		
04	B4	04	B0	00023	CMPC3	(R0), @4(R0), @4(R4)		1369
		36	11	00029	BRB	6\$		
09		52	E9	0002B	1\$:	BLBC	R2, 2\$	1390
52		60	3C	0002E	MOVZWL	(R0), S1_LENGTH		
53	04	A0	D0	00031	MOVL	4(R0), ST_ADDR		
		09	11	00035	BRB	3\$		
		00	16	00037	2\$:	JSB	STR\$ANALYZE_SDESC_R1	
52	00000000G	50	7D	0003D	MOVQ	R0, R2		
02		A4	91	00040	3\$:	CMPB	3(R4), #2	1391
		09	1A	00044	BGTRU	4\$		
50		64	3C	00046	MOVZWL	(R4), S2_LENGTH		
51	04	A4	D0	00049	MOVL	4(R4), S2_ADDR		
		09	11	0004D	BRB	5\$		
50	00000000G	54	D0	0004F	4\$:	MOVL	R4, R0	
		00	16	00052	JSB	STR\$ANALYZE_SDESC_R1		
50		52	D1	00058	5\$:	CMPL	S1_LENGTH, S2_LENGTH	1398
		0A	12	0005B	BNEQ	7\$		
61		52	29	0005D	6\$:	CMPC3	S1_LENGTH, (S1_ADDR), (S2_ADDR)	1401
		04	12	00061	BNEQ	7\$		
		55	D4	00063	CLRL	MATCH		1407
		03	11	00065	BRB	8\$		
55		01	D0	00067	7\$:	MOVL	#1, MATCH	1410
50		55	D0	0006A	8\$:	MOVL	MATCH, R0	1414
		04	0006D		RET			1415

; Routine Size: 110 bytes, Routine Base: _STR\$CODE + 0000

STR\$COMPARE_EQL
1-005

D 2
16-Sep-1984 01:32:27
14-Sep-1984 12:40:01

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

Page 7
(4)

: 226 1416 1 END
: 227 1417 0 ELUDOM

!End of module

PSECT SUMMARY

Name Bytes Attributes
:_STR\$CODE 110 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	4	0	581	00:00.8

COMMAND QUALIFIERS

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

: Size: 110 code + 0 data bytes
: Run Time: 00:05.6
: Elapsed Time: 00:17.5
: Lines/CPU Min: 15291
: Lexemes/CPU-Min: 37834
: Memory Used: 91 pages
: Compilation Complete

