

[illegible]

Sy

LI

LI
LILI
LILI
LILI
LILI
LI

LI

LI

LI

LI
LILI
LILI
LI

11

LI
LI

LI

LI

21

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

LI
LI

11

100

10

111

LI

LI

LI
LI

11

10

1000

10

10

1

1

10

STAR
1-C

: R

[illegible]


```
0001 0 MODULE STR$GET_FREE (
0002 0 IDENT = '1-012'
0003 0 ) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *
0027 1 *****
0028 1
0029 1
0030 1 ++
0031 1 FACILITY: String handling : allocation.
0032 1
0033 1 ABSTRACT:
0034 1
0035 1 This module contains routines to get and free strings. Within
0036 1 the STR$ facility these operations are performed in-line;
0037 1 these routines are for outsiders who wish to allocate and
0038 1 deallocate strings.
0039 1
0040 1 ENVIRONMENT: VAX-11 User Mode
0041 1
0042 1 AUTHOR: John Sauter, CREATION DATE: 14-MAR-1979
0043 1
0044 1 MODIFIED BY:
0045 1
0046 1 1-001 - Original. JBS 15-MAR-1979
0047 1 1-002 - Use the new STR error messages. Conversion of input scalars
0048 1 to by-reference will wait until later. JBS 16-MAY-1979
0049 1 1-003 - Convert the one input scalar to by-reference. JBS 18-MAY-1979
0050 1 1-004 - Add JSB entry points. JBS 22-MAY-1979
0051 1 1-005 - Make the JSB entry points end in _R4. JBS 22-MAY-1979
0052 1 1-006 - Use STRLNK.REQ. JBS 04-JUN-1979
0053 1 1-007 - Make some minor edits based on the code review. JBS 12-JUN-1979
0054 1 1-008 - Correct some misspelled comments. JBS 22-JUN-1979
0055 1 1-009 - Improve some comments. JBS 30-JUL-1979
0056 1 1-010 - String speedup, status from alloc and dealloc macros. RW 11-Jan-1980
0057 1 1-011 - Correct a typo in STR$GET1_DX_R4 (an extra dot). JBS 04-MAR-1980
```

! Allocate and deallocate dynamic strings
! File: STRGETFRE.B32 Edit: LEB1012

STR\$GET_FREE
1-012

C 10
16-Sep-1984 01:39:31
14-Sep-1984 12:40:07

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

Page 2
(1)

```

: 58      0058 1 ! 1-012 - Change EXTERNAL ROUTINE declaration of LIB$STOP so that it matches
: 59      0059 1 ! the way it is declared in macro $STR$SIGNAL_FATAL (without the
: 60      0060 1 ! NOVALUE attribute). This is fallout from a new BLISS compiler.
: 61      0061 1 !
: 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 (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
71 0070 1 |
72 0071 1 |
73 0072 1 | LINKAGES:
74 0073 1 |
75 0074 1 |
76 0075 1 | REQUIRE 'RTLIN:STRLNK';           ! JSB linkages
77 0260 1 |
78 0261 1 |
79 0262 1 | TABLE OF CONTENTS:
80 0263 1 |
81 0264 1 |
82 0265 1 | FORWARD ROUTINE
83 0266 1 |     STR$GET1_DX,                 ! Allocate a string
84 0267 1 |     STR$GET1_DX_R4 : STR$JSB_GETFRE, ! (JSB entry point)
85 0268 1 |     STR$FREE1_DX,               ! Deallocate a string
86 0269 1 |     STR$FREE1_DX_R4 : STR$JSB_GETFRE; ! (JSB entry point)
87 0270 1 |
88 0271 1 |
89 0272 1 | INCLUDE FILES:
90 0273 1 |
91 0274 1 |
92 0275 1 | REQUIRE 'RTLIN:RTLPSECT';         ! Macros for defining psects
93 0370 1 |
94 0371 1 | REQUIRE 'RTLIN:STRMACROS';        ! String macros
95 1287 1 |
96 1288 1 | LIBRARY 'RTLSTARLE';             ! System symbols
97 1289 1 |
98 1290 1 |
99 1291 1 | MACROS:
100 1292 1 |
101 1293 1 |     NONE
102 1294 1 |
103 1295 1 | EQUATED SYMBOLS:
104 1296 1 |
105 1297 1 |     NONE
106 1298 1 |
107 1299 1 | PSECTS:
108 1300 1 |
109 1301 1 | DECLARE_PSECTS (STR);             ! Declare psects for STR$ facility
110 1302 1 |
111 1303 1 | OWN STORAGE:
112 1304 1 |
113 1305 1 |     NONE
114 1306 1 |
115 1307 1 | EXTERNAL REFERENCES:
116 1308 1 |
117 1309 1 |
118 1310 1 | EXTERNAL ROUTINE
119 1311 1 |     LIB$STOP ;                   ! Signal a fatal error
120 1312 1 |
121 1313 1 |
122 1314 1 | +
    | The following are the error codes used by this module.
```

STR\$GET_FREE
1-012

E 10
16-Sep-1984 01:39:31
14-Sep-1984 12:40:07

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

Page 4
(2)

```
: 123      1315 1 !-
: 124      1316 1
: 125      1317 1 EXTERNAL LITERAL
: 126      1318 1 STR$NORMAL : UNSIGNED (6),
: 127      1319 1 STR$_ILLSTRCLA;
: 128      1320 1
```

```
! Success
! Illegal string class
```



```
130 1321 1 GLOBAL ROUTINE STR$GET1_DX (
131 1322 1     LEN,
132 1323 1     DESCRIP
133 1324 1 ) =
134 1325 1
135 1326 1 ++
136 1327 1 FUNCTIONAL DESCRIPTION:
137 1328 1
138 1329 1     Allocate a string. LEN bytes are allocated to DESCRIP, which
139 1330 1     must be a dynamic descriptor. If the descriptor already
140 1331 1     has enough storage allocated to it, nothing is done. However,
141 1332 1     if the descriptor does not reference a "short string", it cannot
142 1333 1     be shortened: it will be deallocated and reallocated instead.
143 1334 1
144 1335 1 FORMAL PARAMETERS:
145 1336 1
146 1337 1     LEN.rw.r      Number of bytes to allocate in the string.
147 1338 1     DESCRIP.mq.r  The descriptor which will be re-allocated.
148 1339 1
149 1340 1 IMPLICIT INPUTS:
150 1341 1
151 1342 1     NONE
152 1343 1
153 1344 1 IMPLICIT OUTPUTS:
154 1345 1
155 1346 1     NONE
156 1347 1
157 1348 1 COMPLETION CODES:
158 1349 1
159 1350 1     Always $$$_NORMAL
160 1351 1
161 1352 1 SIDE EFFECTS:
162 1353 1
163 1354 1     May deallocate the descriptor's storage and allocate new
164 1355 1     storage for it.
165 1356 1     Signals ILLEGAL STRING CLASS if the descriptor is not dynamic.
166 1357 1     Signals STR$_INSVIRMEM if can't get dynamic string space
167 1358 1     Signals STR$_FATINTERR if can't free space allocated to string
168 1359 1
169 1360 1 --
170 1361 1 BEGIN
171 1362 2
172 1363 2 MAP
173 1364 2     DESCRIP : REF $STR$DESCRIPTOR;
174 1365 2
175 1366 2 ++
176 1367 2 Verify that the descriptor is a dynamic string.
177 1368 2
178 1369 2 --
179 1370 2
180 1371 2 CASE $STR$CLASS (DESCRIP) FROM DSC$K_CLASS_Z TO DSC$K_CLASS_D OF
181 1372 2 SET
182 1373 2
183 1374 2     [DSC$K_CLASS_D] :
184 1375 2         BEGIN
185 1376 3
186 1377 3         LOCAL
```

```
187      RETURN_STATUS;                ! remember status
188
189      !+ Prevent AST's to change string while we are looking at it. If string is
190      !- already being written, signal an error
191      $STR$INTERLOCK (1);
192
193      IF (RETURN_STATUS = $STR$INTERLOCK_WRITE (.DESCRIP, 0))
194      THEN
195      !+ This is a dynamic descriptor. If it has enough storage already,
196      !- we need not re-allocate it.
197
198      IF ($STR$NEED_ALLOC ((..LEN AND %X'FFFF'), $STR$DYN_AL_LEN (DESCRIP)))
199      THEN
200      BEGIN
201      !+ We must deallocate and reallocate the string. If the deallocate
202      !- succeeds then do the reallocate.
203
204      IF (RETURN_STATUS = $STR$DEALLOCATE (DESCRIP))
205      THEN
206      RETURN_STATUS = $STR$ALLOCATE ((..LEN AND %X'FFFF'), DESCRIP);
207
208      END
209      ELSE
210      $STR$LENGTH (DESCRIP) = (..LEN AND %X'FFFF');
211
212      $STR$INTERLOCK_CLEAR (.DESCRIP, 0);
213      $STR$SIGNAL_FATAL (RETURN_STATUS);
214      END;
215
216      [DSC$K_CLASS_S, INRANGE, OUTRANGE] :
217      LIB$STOP-(STR$_ILLSTRCLA);
218      TES;
219
220      RETURN (SS$_NORMAL);
221      END;
222
223      ! end of STR$GET1_DX
```

```
.TITLE STR$GET_FREE
.IDENT \1-012\
```

```
.EXTRN LIB$STOP, STR$_NORMAL
.EXTRN STR$_ILLSTRCLA, STR$_SHORT_Q
.EXTRN LIB$FREE_VM, STR$_FATINTERR
.EXTRN STR$_INIT, STR$_SV_INIT
.EXTRN STR$_ALLOC_SHORT
.EXTRN LIB$GET_VM, STR$_INSVIRMEM
```

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

```
.ENTRY STR$GET1_DX, Save R2, R3, R4, R5, R6
MOVAB STR$_SHORT_Q, R6
```

```
007C 00000
56 00000000G 00 9E 00002
```

```
: 1321
:
```


02	5E	08	04	C2	00009	SUBL2	#4, SP		
000F	52	03	AC	D0	0000C	MOVL	DESCRIP, R2		1371
	00		A2	8F	00010	CASEB	3(R2), #0, #2		
	0006		0006		00015	.WORD	2\$-1\$,-		
							2\$-1\$,-		
							3\$-1\$		
							#STR\$_ILLSTRCLA		1414
							30\$		
	55						#1, RETURN_STATUS		1386
	53	04	A2	D0	00027	MOVL	4(R2), R3		1393
			54	D4	0002B	CLRL	R4		
			53	D5	0002D	TSTL	R3		
			06	12	0002F	BNEQ	4\$		
			54	D6	00031	INCL	R4		
			50	D4	00033	CLRL	R0		
			13	11	00035	BRB	6\$		
00F0	8F		62	B1	00037	CMPW	(R2), #240		
			05	1B	0003C	BLEQU	5\$		
	50		62	3C	0003E	MOVZWL	(R2), R0		
			07	11	00041	BRB	6\$		
	50		53	D0	00043	MOVL	R3, STRING_BLOCK		
	50	FE	A0	3C	00046	MOVZWL	-2(STRING_BLOCK), R0		
000000F0	8F		50	D1	0004A	CMPL	R0, #240		
			25	1F	00051	BLSSU	10\$		
	51	04	BC	3C	00053	MOVZWL	@LEN, R1		
	04		54	E9	00057	BLBC	R4, 7\$		
			50	D4	0005A	CLRL	R0		
			13	11	0005C	BRB	9\$		
00F0	8F		62	B1	0005E	CMPW	(R2), #240		
			05	1B	00063	BLEQU	8\$		
	50		62	3C	00065	MOVZWL	(R2), R0		
			07	11	00068	BRB	9\$		
	50		53	D0	0006A	MOVL	R3, STRING_BLOCK		
	50	FE	A0	3C	0006D	MOVZWL	-2(STRING_BLOCK), R0		
	50		51	D1	00071	CMPL	R1, R0		
			25	13	00074	BEQL	14\$		
			26	11	00076	BRB	15\$		
	51	04	BC	3C	00078	MOVZWL	@LEN, R1		
	04		54	E9	0007C	BLBC	R4, 11\$		
			50	D4	0007F	CLRL	R0		
			13	11	00081	BRB	13\$		
00F0	8F		62	B1	00083	CMPW	(R2), #240		
			05	1B	00088	BLEQU	12\$		
	50		62	3C	0008A	MOVZWL	(R2), R0		
			07	11	0008D	BRB	13\$		
	50		53	D0	0008F	MOVL	R3, STRING_BLOCK		
	50	FE	A0	3C	00092	MOVZWL	-2(STRING_BLOCK), R0		
	50		51	D1	00096	CMPL	R1, R0		
			03	1A	00099	BGTRU	15\$		
			00D1	31	0009B	BRW	28\$		
	50		00G	9A	0009E	MOVZBL	S^STR\$_NORMAL, RETURN_STATUS		1401
			53	D5	000A1	TSTL	R3		
			37	13	000A3	BEQL	17\$		
00F0	8F		62	B1	000A5	CMPW	(R2), #240		
			15	1A	000AA	BGTRU	16\$		
	51		53	D0	000AC	MOVL	R3, STRING_BLOCK		
	51	FE	A1	3C	000AF	MOVZWL	-2(STRING_BLOCK), ALLOC_LENGTH		

			51	D7	000B3	DECL	R1		
			07	8A	000B5	BICB2	#7, R1		
			56	C0	000B8	ADDL2	R6, R1		
	00		B1	63	0E	000BB	INSQUE	(R3), @0(INSQUE_ADDR)	
				1B	11	000BF	BRB	17\$	
		04		A2	9F	000C1	PUSHAB	4(R2)	
	04	AE		62	3C	000C4	MOVZWL	(R2), 4(SP)	
				AE	9F	000C8	PUSHAB	4(SP)	
00000000G	00		02	FB	000CB	CALLS	#2, LIB\$FREE VM		
	07		50	E8	000D2	BLBS	RETURN_STATUS, 17\$		
	50	00000000G	8F	D0	000D5	MOVL	#STR\$ FATINTERR, RETURN_STATUS		
	55		50	D0	000DC	MOVL	RETURN_STATUS, RETURN_STATUS		
	03		50	E8	000DF	BLBS	RETURN_STATUS, 18\$		
			08D	31	000E2	BRW	29\$		
00000000G	07	00000000G	00	E8	000E5	BLBS	STR\$SV INIT, 19\$		1403
	00		00	FB	000EC	CALLS	#0, STR\$INIT		
	50		00G	9A	000F3	MOVZBL	S^STR\$ NORMAL, RETURN_STATUS		
00F0	8F		04	BC	B1	000F6	CMPW	@LEN, #240	
				47	1A	000FC	BGTRU	25\$	
			04	BC	B5	000FE	TSTW	@LEN	
			04	12	00101	BNEQ	20\$		
			53	D4	00103	CLRL	TEMP		
			2D	11	00105	BRB	24\$		
	51		04	BC	3C	00107	MOVZWL	@LEN, R1	
				51	D7	0010B	DECL	R1	
54	51		07	8A	0010D	BICB2	#7, R1		
	51		56	C1	00110	ADDL3	R6, R1, REMQUE_ADDR		
	53		00	B4	0F	00114	REMQUE	@0(REMQUE_ADDR), TEMP	
			05	1D	00118	BVS	22\$		
	52		01	D0	0011A	MOVL	#1, ALLOC_DONE		
			0D	11	0011D	BRB	23\$		
			52	D4	0011F	CLRL	ALLOC_DONE		
			04	BC	3C	00121	MOVZWL	@LEN, -(SP)	
00000000G	7E		01	FB	00125	CALLS	#1, STR\$ALLOC SHORT		
	00		52	E8	0012C	BLBS	ALLOC_DONE, 24\$		
	05		50	E9	0012F	BLBC	RETURN_STATUS, 27\$		
	38		E0	11	00132	BRB	21\$		
			50	E9	00134	BLBC	RETURN_STATUS, 27\$		
	33		08	AC	D0	00137	MOVL	DESCRIP, R1	
	04	A1	53	D0	0013B	MOVL	TEMP, 4(R1)		
		61	04	BC	B0	0013F	MOVW	@LEN, (R1)	
			25	11	00143	BRB	27\$		
7E	08	AC	04	C1	00145	ADDL3	#4, DESCRIP, -(SP)		
	04	AE	04	BC	3C	0014A	MOVZWL	@LEN, 4(SP)	
			04	AE	9F	0014F	PUSHAB	4(SP)	
00000000G	00		02	FB	00152	CALLS	#2, LIB\$GET VM		
	09		50	E8	00159	BLBS	RETURN_STATUS, 26\$		
	50	00000000G	8F	D0	0015C	MOVL	#STR\$ INSVIRMEM, RETURN_STATUS		
			05	11	00163	BRB	27\$		
	08	BC	04	BC	B0	00165	MOVW	@LEN, @DESCRIP	
		55	50	D0	0016A	MOVL	RETURN_STATUS, RETURN_STATUS		
			03	11	0016D	BRB	29\$		
	62		51	B0	0016F	MOVW	R1, (R2)		1393
	10		55	E8	00172	BLBS	RETURN_STATUS, 31\$		1407
04		55	00	ED	00175	CMPZV	#0, #3, RETURN_STATUS, #4		1410
			09	12	0017A	BNEQ	31\$		
			55	DD	0017C	PUSHL	RETURN_STATUS		

STR\$GET_FREE
1-012

J 10
16-Sep-1984 01:39:31 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:07 [LIBRTL.SRC]STRGETFRE.B32;1

Page 9
(3)

00000000G 00
50

01 FB 0017E 30\$: CALLS #1, LIB\$STOP
01 DO 00185 31\$: MOVL #1, R0
04 00188 RET

: 1417
: 1418

; Routine Size: 393 bytes, Routine Base: _STR\$CODE + 0000

; 228 1419 1

```
230 1420 1 GLOBAL ROUTINE STR$GET1_DX_R4 (
231 1421 1     LEN,
232 1422 1     DESCRIP
233 1423 1 ) : STR$JSB_GETFRE =
234 1424 1
235 1425 1 ++
236 1426 1 FUNCTIONAL DESCRIPTION:
237 1427 1
238 1428 1     Allocate a string. LEN bytes are allocated to DESCRIP, which
239 1429 1     had better be a dynamic descriptor. If the descriptor already
240 1430 1     has enough storage allocated to it, nothing is done. However,
241 1431 1     if the descriptor does not reference a "short string", it cannot
242 1432 1     be shortened: it will be deallocated and reallocated instead.
243 1433 1
244 1434 1 FORMAL PARAMETERS:
245 1435 1
246 1436 1     LEN.rw.v      Number of bytes to allocate in the string.
247 1437 1     DESCRIP.mq.r  The descriptor which will be re-allocated.
248 1438 1
249 1439 1 IMPLICIT INPUTS:
250 1440 1
251 1441 1     NONE
252 1442 1
253 1443 1 IMPLICIT OUTPUTS:
254 1444 1
255 1445 1     NONE
256 1446 1
257 1447 1 COMPLETION CODES:
258 1448 1
259 1449 1     Always $$$_NORMAL
260 1450 1
261 1451 1 SIDE EFFECTS:
262 1452 1
263 1453 1     May deallocate the descriptor's storage and allocate new
264 1454 1     storage for it.
265 1455 1     Signals ILLEGAL STRING CLASS if the descriptor is not dynamic.
266 1456 1     Signals STR$_INSVIRMEM if can't get dynamic string space
267 1457 1     Signals STR$_FATINTERR if can't free space allocated to string
268 1458 1
269 1459 1 --
270 1460 1
271 1461 2 BEGIN
272 1462 2
273 1463 2 MAP
274 1464 2     DESCRIP : REF $STR$DESCRIPTOR;
275 1465 2
276 1466 2 ++
277 1467 2 Verify that the descriptor is a dynamic string.
278 1468 2 --
279 1469 2
280 1470 2 CASE $STR$CLASS (DESCRIP) FROM DSC$K_CLASS_Z TO DSC$K_CLASS_D OF
281 1471 2 SET
282 1472 2
283 1473 2     [DSC$K_CLASS_D] :
284 1474 2         BEGIN
285 1475 2
286 1476 2         LOCAL
```



```
287      RETURN_STATUS;                ! remember status
288
289
290      + Prevent AST's to change string while we are looking at it.  If string is
291      + already being written, signal an error
292      -
293      $STR$INTERLOCK (1);
294
295      IF (RETURN_STATUS = $STR$INTERLOCK_WRITE (.DESCRIP, 0))
296      THEN
297      + This is a dynamic descriptor.  If it has enough storage already,
298      + we need not re-allocate it.
299      -
300
301      IF ($STR$NEED_ALLOC ((.LEN AND %X'FFFF'), $STR$DYN_AL_LEN (DESCRIP)))
302      THEN
303      BEGIN
304      + We must deallocate and reallocate the string.  If the deallocate
305      + succeeds then do the reallocate.
306      -
307
308      IF (RETURN_STATUS = $STR$DEALLOCATE (DESCRIP))
309      THEN
310      RETURN_STATUS = $STR$ALLOCATE ((.LEN AND %X'FFFF'), DESCRIP);
311
312      END
313      ELSE
314      $STR$LENGTH (DESCRIP) = (.LEN AND %X'FFFF');
315
316      $STR$INTERLOCK_CLEAR (.DESCRIP, 0);
317      $STR$SIGNAL_FATAL (RETURN_STATUS);
318      END;
319
320      [DSC$K_CLASS_S, INRANGE, OUTRANGE] :
321      LIB$STOP-(STR$_ILLSTRCLA);
322      TES;
323
324      RETURN (SS$_NORMAL);
325      END;
326
327
```

! end of STR\$GET1_DX_R4

5E	OC	C2	00000	STR\$GET1	DX R4::		
52	51	D0	00003	SUBC2	#12, SP		1420
54	50	D0	00006	MOVL	R1, R2		
02	03	A2	8F	00009	MOVL	R0, R4	
000F	0006	0006	0000E	1\$:	CASEB	3(DESCRIP), #0, #2	1470
					.WORD	2\$-1\$,-	
						2\$-1\$,-	
						3\$-1\$	
	00000000G	8F	DD	00014	2\$:	PUSHL	#STR\$_ILLSTRCLA
		0159	31	0001A		BRW	30\$
08	AE	01	D0	0001D	3\$:	MOVL	#1, RETURN_STATUS

1485

	53	04	A2	D0	00021	MOVL	4(DESCRIP), R3		1492
			50	D4	00025	CLRL	R0		
			53	D5	00027	TSTL	R3		
			06	12	00029	BNEQ	4\$		
			50	D6	0002B	INCL	R0		
			51	D4	0002D	CLRL	R1		
			13	11	0002F	BRB	6\$		
	00F0	8F	62	B1	00031	4\$: CMPW	(DESCRIP), #240		
			05	1B	00036	BLEQU	5\$		
	51		62	3C	00038	MOVZWL	(DESCRIP), R1		
			07	11	0003B	BRB	6\$		
	51		53	D0	0003D	5\$: MOVL	R3, STRING BLOCK		
	51		A1	3C	00040	MOVZWL	-2(STRING_BLOCK), R1		
	000000F0	8F	51	D1	00044	6\$: CMPL	R1, #240		
			23	1F	0004B	BLSSU	10\$		
	04		50	E9	0004D	BLBC	R0, 7\$		
			51	D4	00050	CLRL	R1		
			13	11	00052	BRB	9\$		
	00F0	8F	62	B1	00054	7\$: CMPW	(DESCRIP), #240		
			05	1B	00059	BLEQU	8\$		
	51		62	3C	0005B	MOVZWL	(DESCRIP), R1		
			07	11	0005E	BRB	9\$		
	51		53	D0	00060	8\$: MOVL	R3, STRING BLOCK		
	51		A1	3C	00063	MCVZWL	-2(STRING_BLOCK), R1		
51	54	10	00	ED	00067	9\$: CMPZV	#0, #16, LEN, R1		
			23	13	0006C	BEQL	14\$		
			24	11	0006E	BRB	15\$		
	04		50	E9	00070	10\$: BLBC	R0, 11\$		
			51	D4	00073	CLRL	R1		
			13	11	00075	BRB	13\$		
	00F0	8F	62	B1	00077	11\$: CMPW	(DESCRIP), #240		
			05	1B	0007C	BLEQU	12\$		
	51		62	3C	0007E	MOVZWL	(DESCRIP), R1		
			07	11	00081	BRB	13\$		
	51		53	D0	00083	12\$: MOVL	R3, STRING BLOCK		
	51		A1	3C	00086	MOVZWL	-2(STRING_BLOCK), R1		
51	54	10	00	ED	0008A	13\$: CMPZV	#0, #16, LEN, R1		
			03	1A	0008F	BGTRU	15\$		
			00D0	31	00091	BRW	28\$		
	50		00G	9A	00094	15\$: MOVZBL	S^STR\$_NORMAL, RETURN_STATUS		1500
			53	D5	00097	TSTL	R3		
			3C	13	00099	BEQL	17\$		
	00F0	8F	62	B1	0009B	CMPW	(DESCRIP), #240		
			1A	1A	000A0	BGTRU	16\$		
	51		53	D0	000A2	MOVL	R3, STRING BLOCK		
	51		A1	3C	000A5	MOVZWL	-2(STRING_BLOCK), ALLOC_LENGTH		
			51	D7	000A9	DECL	R1		
	51		07	8A	000AB	BICB2	#7, R1		
	51	00000000G00	41	9E	000AE	MOVAB	STR\$Q SHORT Q[R1], INSQUE_ADDR		
	00	B1	63	0E	000B6	INSQUE	(R3), #0(INSQUE_ADDR)		
			1B	11	000BA	BRB	17\$		
			04	A2	9F	000BC	16\$: PUSHAB	4(DESCRIP)	
	08	AE	62	3C	000BF	MOVZWL	(DESCRIP), 8(SP)		
			08	AE	9F	000C3	PUSHAB	8(SP)	
	00000000G	00	02	FB	000C6	CALLS	#2, LIB\$FREE VM		
		07	50	E8	000CD	BLBS	RETURN STATUS, 17\$		
	50	00000000G	8F	D0	000D0	MOVL	#STR\$_FATINTERR, RETURN_STATUS		

STR\$GET_FREE
1-012

N 10

16-Sep-1984 01:39:31

14-Sep-1984 12:40:07

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

Page 13
(4)

08	AE	50	D0	000D7	17\$:	MOVL	RETURN_STATUS, RETURN_STATUS	
	03	50	E8	000DB		BLBS	RETURN_STATUS, 18\$	
		0086	31	000DE		BRW	29\$	
00000000G	07	00000000G	00	E8	000E1	18\$:	BLBS	STR\$V_INIT, 19\$
	00		00	FB	000E8		CALLS	#0, STR\$INIT
00F0	50	00G	9A	000EF	19\$:	MOVZBL	S^STR\$ NORMAL, RETURN_STATUS	
	8F	54	B1	000F2		CMPW	LEN, #240	
		45	1A	000F7		BGTRU	25\$	
		54	B5	000F9		TSTW	LEN	
		04	12	000FB		BNEQ	20\$	
		6E	D4	000FD		CLRL	TEMP	
		34	11	000FF		BRB	24\$	
	51	54	3C	00101	20\$:	MOVZWL	LEN, R1	
		51	D7	00104		DECL	R1	
	51	07	8A	00106		BICB2	#7, R1	
04	AE	00000000G0041	9E	00109		MOVAB	STR\$Q_SHORT_Q[R1], REMQUE_ADDR	
	51	04	BE	9E	00112	21\$:	MOVAB	@REMQUE_ADDR, R1
	6E	00	B1	0F	00116		REMQUE	@0(R1), TEMP
		05	1D	0011A		BVS	22\$	
	53	01	D0	0011C		MOVL	#1, ALLOC_DONE	
		0C	11	0011F		BRB	23\$	
		53	D4	00121	22\$:	CLRL	ALLOC_DONE	
00000000G	7E	54	3C	00123		MOVZWL	LEN, =(SP)	
	00	01	FB	00126		CALLS	#1, STR\$ALLOC_SHORT	
	05	53	E8	0012D	23\$:	BLBS	ALLOC_DONE, 24\$	
	2B	50	E9	00130		BLBC	RETURN_STATUS, 27\$	
		DD	11	00133		BRB	21\$	
	26	50	E9	00135	24\$:	BLBC	RETURN_STATUS, 27\$	
04	A2	6E	D0	00138		MOVL	TEMP, 4(DESCRIPT)	
		1D	11	0013C		BRB	26\$	
		04	A2	9F	0013E	25\$:	PUSHAB	4(DESCRIPT)
08	AE	54	3C	00141		MOVZWL	LEN, 8(SP)	
		08	AE	9F	00145	PUSHAB	8(SP)	
00000000G	00	02	FB	00148		CALLS	#2, LIB\$GET_VM	
	09	50	E8	0014F		BLBS	RETURN_STATUS, 26\$	
	50	00000000G	8F	D0	00152	MOV	#STR\$_INSVIRMEM, RETURN_STATUS	
		03	11	00159		BRB	27\$	
	62	54	B0	0015B	26\$:	MOVW	LEN, (DESCRIPT)	
08	AE	50	D0	0015E	27\$:	MOVL	RETURN_STATUS, RETURN_STATUS	
		03	11	00162		BRB	29\$	
	62	54	B0	00164	28\$:	MOVW	LEN, (DESCRIPT)	
	12	08	AE	E8	00167	29\$:	BLBS	RETURN_STATUS, 31\$
04	08	AE	00	ED	0016B		CMPZV	#0, #3, RETURN_STATUS, #4
	03	0A	12	00171		BNEQ	31\$	
		08	AE	DD	00173	PUSHL	RETURN_STATUS	
00000000G	00	01	FB	00176	30\$:	CALLS	#1, LIB\$STOP	
	50	01	D0	0017D	31\$:	MOVL	#1, R0	
	5E	0C	C0	00180		ADDL2	#12, SP	
		05	00183		RSB			

; Routine Size: 388 bytes, Routine Base: _STR\$CODE + 0189

; 328 1518 1

```
330 1519 1 GLOBAL ROUTINE STR$FREE1_DX (      ! Deallocate a string
331 1520 1     DESCRIP                          ! The descriptor to deallocate
332 1521 1     ) =
333 1522 1
334 1523 1 !++
335 1524 1 FUNCTIONAL DESCRIPTION:
336 1525 1
337 1526 1     Deallocate a string. The string must be dynamic. If the
338 1527 1     string is "short", it is put on a queue of strings of the
339 1528 1     proper length. Otherwise it is returned to virtual memory.
340 1529 1
341 1530 1 FORMAL PARAMETERS:
342 1531 1
343 1532 1     DESCRIP.wq.r    The descriptor of the string to deallocate.
344 1533 1
345 1534 1 IMPLICIT INPUTS:
346 1535 1
347 1536 1     NONE
348 1537 1
349 1538 1 IMPLICIT OUTPUTS:
350 1539 1
351 1540 1     NONE
352 1541 1
353 1542 1 ROUTINE VALUE:
354 1543 1 COMPLETION CODES:
355 1544 1
356 1545 1     Always SS$_NORMAL
357 1546 1
358 1547 1 SIDE EFFECTS:
359 1548 1
360 1549 1     May deallocate virtual storage.
361 1550 1     May signal STR$_FATINTERR if can't deallocate string
362 1551 1     May signal STR$_ILLSTRCLA if not a dynamic string
363 1552 1
364 1553 1 --
365 1554 1
366 1555 2 BEGIN
367 1556 2
368 1557 2 MAP
369 1558 2     DESCRIP : REF $STR$DESCRIPTOR;
370 1559 2
371 1560 2 !+
372 1561 2 Verify that this is a dynamic descriptor.
373 1562 2 !-
374 1563 2
375 1564 2 CASE $STR$CLASS (DESCRIP) FROM DSC$_K_CLASS_S TO DSC$_K_CLASS_D OF
376 1565 2     SET
377 1566 2
378 1567 2     [DSC$_K_CLASS_D] :
379 1568 2         BEGIN
380 1569 2
381 1570 2         LOCAL
382 1571 2             RETURN_STATUS;      ! remember status
383 1572 2
384 1573 2 !+
385 1574 2 Prevent AST's to change string while we are looking at it. If string is
386 1575 2 already being written, signal an error
```



```
387 1576 3 :-
388 1577 $STR$INTERLOCK (1);
389 1578
390 1579 IF (RETURN_STATUS = $STR$INTERLOCK_WRITE (.DESCRIP, 0))
391 1580 THEN
392 1581 BEGIN
393 1582
394 1583 + Deallocate the string data.
395 1584 -
396 1585
397 1586 IF (RETURN_STATUS = $STR$DEALLOCATE (DESCRIP))
398 1587 +
399 1588 Make sure the pointer and length field are zero, so the user is less
400 1589 likely to mistakenly use an old address. Also, if he calls
401 1590 STR$GET1_DX without reinitializing the descriptor it will not get
402 1591 confused.
403 1592 -
404 1593 THEN
405 1594 BEGIN
406 1595 DESCRIP [DSC$W_LENGTH] = 0;
407 1596 DESCRIP [DSC$A_POINTER] = 0;
408 1597 END;
409 1598
410 1599 END;
411 1600
412 1601 $STR$INTERLOCK_CLEAR (.DESCRIP, 0);
413 1602 $STR$SIGNAL_FATAL (RETURN_STATUS);
414 1603 END;
415 1604
416 1605 [INRANGE, OTRANGE] :
417 1606 LIB$STOP (STR$_ILLSTRCLA);
418 1607 TES;
419 1608
420 1609 RETURN (SS$_NORMAL);
421 1610 END;

! end of STR$FREE1_DX
```

01	5E	04	001C	00000	.ENTRY	STR\$FREE1_DX, Save R2,R3,R4	1519
	52	04	C2	00002	SUBL2	#4, SP	
	01	03	AC	D0 00005	MOVL	DESCRIP, R2	1564
	000C	0004	A2	8F 00009	CASEB	3(R2), #1, #1	
				0000E 1\$:	.WORD	2\$-1\$,-	
						3\$-1\$	
		00000000G	8F	DD 00012 2\$:	PUSHL	#STR\$_ILLSTRCLA	1606
			63	11 00018	BRB	7\$	
	54		01	D0 0001A 3\$:	MOVL	#1, RETURN STATUS	1579
	50		00G	9A 0001D	MOVZBL	S^STR\$_NORMAL, RETURN_STATUS	1586
	53	04	A2	D0 00020	MOVL	4(R2), R3	
			3C	13 00024	BEQL	5\$	
00F0	8F		62	B1 00026	CMPL	(R2), #240	
			1A	1A 0002B	BGTRU	4\$	
	51		53	D0 0002D	MOVL	R3, STRING_BLOCK	
	51	FE	A1	3C 00030	MOVZWL	-2(STRING_BLOCK), ALLOC_LENGTH	
			51	D7 00034	DECL	R1	

STR\$GET_FREE
1-012

D 11
16-Sep-1984 01:39:31
14-Sep-1984 12:40:07

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

Page 16
(5)

	51		07	8A	00036	BICB2	#7, R1	:
	51	00000000G00	41	9E	00039	MOVAB	STR\$Q SHORT Q[R1], INSQUE_ADDR	:
00	B1		63	0E	00041	INSQUE	(R3), 30(INSQUE_ADDR)	:
			1B	11	00045	BRB	5\$	:
		04	A2	9F	00047	PUSHAB	4(R2)	:
04	AE		62	3C	0004A	MOVZWL	(R2), 4(SP)	:
		04	AE	9F	0004E	PUSHAB	4(SP)	:
00000000G	00		02	FB	00051	CALLS	#2, LIB\$FREE VM	:
	07		50	E8	00058	BLBS	RETURN_STATUS, 5\$	:
	50	00000000G	8F	D0	0005B	MOVL	#STR\$ FATINTERR, RETURN_STATUS	:
	54		50	D0	00062	MOVL	RETURN_STATUS, RETURN_STATUS	:
	09		50	E9	00065	BLBC	RETURN_STATUS, 6\$	:
	50		04	AC	00068	MOVL	DESCRIP, R0	1595
			60	B4	0006C	CLRW	(R0)	:
			04	A0	0006E	CLRL	4(R0)	1596
	10		54	E8	00071	BLBS	RETURN_STATUS, 8\$	1602
04	03		00	ED	00074	CMPZV	#0, #3, RETURN_STATUS, #4	:
			09	12	00079	BNEQ	8\$	:
			54	DD	0007B	PUSHL	RETURN_STATUS	:
00000000G	00		01	FB	0007D	CALLS	#1, LIB\$STOP	:
	50		01	D0	00084	MOVL	#1, R0	1609
			04	00087	RET			1610

; Routine Size: 136 bytes, Routine Base: _STR\$CODE + 030D

; 422 1611 1


```

424 1612 1 GLOBAL ROUTINE STR$FREE1_DX_R4 (           ! Deallocate a string
425 1613 1     DESCRIP                               ! The descriptor to deallocate
426 1614 1     ) : STR$JSB_GETFRE =
427 1615 1
428 1616 1 ++
429 1617 1 FUNCTIONAL DESCRIPTION:
430 1618 1
431 1619 1     Deallocate a string. The string must be dynamic. If the
432 1620 1     string is 'short', it is put on a queue of strings of the
433 1621 1     proper length. Otherwise it is returned to virtual memory.
434 1622 1
435 1623 1 FORMAL PARAMETERS:
436 1624 1
437 1625 1     DESCRIP.wq.r    The descriptor of the string to deallocate.
438 1626 1
439 1627 1 IMPLICIT INPUTS:
440 1628 1
441 1629 1     NONE
442 1630 1
443 1631 1 IMPLICIT OUTPUTS:
444 1632 1
445 1633 1     NONE
446 1634 1
447 1635 1 COMPLETION CODES:
448 1636 1
449 1637 1     Always SS$_NORMAL.
450 1638 1
451 1639 1 SIDE EFFECTS:
452 1640 1
453 1641 1     May deallocate virtual storage.
454 1642 1     May signal STR$_FATINTERR if can't deallocate string
455 1643 1     May signal STR$_ILLSTRCLA if not a dynamic string
456 1644 1
457 1645 1 --
458 1646 1
459 1647 2 BEGIN
460 1648 2
461 1649 2 MAP
462 1650 2     DESCRIP : REF $STR$DESCRIPTOR;
463 1651 2
464 1652 2 ++
465 1653 2 Verify that this is a dynamic descriptor.
466 1654 2 --
467 1655 2
468 1656 2 CASE $STR$CLASS (DESCRIP) FROM DSC$_K_CLASS_S TO DSC$_K_CLASS_D OF
469 1657 2     SET
470 1658 2
471 1659 2     [DSC$_K_CLASS_D] :
472 1660 2         BEGIN
473 1661 2
474 1662 2         LOCAL
475 1663 2             RETURN_STATUS;           ! remember status
476 1664 2
477 1665 2 ++
478 1666 2 Prevent AST's to change string while we are looking at it. If string is
479 1667 2 already being written, signal an error
480 1668 2 --
```

```

481      1669 3      $STR$INTERLOCK (1);
482      1670 3
483      1671 4      IF (RETURN_STATUS = $STR$INTERLOCK_WRITE (.DESCRIP, 0))
484      1672 3      THEN
485      1673 4          BEGIN
486      1674 4      !+ Deallocate the string data.
487      1675 4      !-
488      1676 4
489      1677 4
490      1678 5          IF (RETURN_STATUS = $STR$DEALLOCATE (DESCRIP))
491      1679 5      !+
492      1680 5      ! Make sure the pointer and length field are zero, so the user is less
493      1681 5      ! likely to mistakenly use an old address. Also, if he calls
494      1682 5      ! STR$GET1_DX without reinitializing the descriptor it will not get
495      1683 5      ! confused.
496      1684 5      !-
497      1685 4          THEN
498      1686 5              BEGIN
499      1687 5                  DESCRIP [DSCSW_LENGTH] = 0;
500      1688 5                  DESCRIP [DSCSA_POINTER] = 0;
501      1689 4              END;
502      1690 4
503      1691 3          END;
504      1692 3
505      1693 3      $STR$INTERLOCK_CLEAR (.DESCRIP, 0);
506      1694 3      $STR$SIGNAL_FATAL (RETURN_STATUS);
507      1695 3      END;
508      1696 2
509      1697 2      [INRANGE, OVRANGE] :
510      1698 2      LIB$STOP (STR$_ILLSTRCLA);
511      1699 2      TES;
512      1700 2
513      1701 2      RETURN (SS$_NORMAL);
514      1702 1      END;

```

! end of STR\$FREE1_DX_R4

SE	04	C2	00000	STR\$FREE1_DX_R4::		
52	50	D0	00003	SUBL2	#4, SP	1612
01	A2	8F	00006	MOVL	R0, R2	
000C	03	0004	0000B	CASEB	3(DESCRIP), #1, #1	1656
				.WORD	2\$-1\$, -	
					3\$-1\$	
	00000000G	8F	DD	0000F	2\$: PUSHL	#STR\$_ILLSTRCLA
		5F	11	00015	BRB	7\$
54	01	D0	00017	3\$: MOVL	#1, RETURN STATUS	1671
50	00G	9A	0001A	MOVZBL	S^STR\$ NORMAL, RETURN_STATUS	1678
53	04	A2	D0	0001D	MOVL	4(DESCRIP), R3
		3C	13	00021	BEQL	5\$
00F0	8F	62	B1	00023	CMPW	(DESCRIP), #240
		1A	1A	00028	BGTRU	4\$
51	53	D0	0002A	MOVL	R3, STRING_BLOCK	
51	FE	A1	3C	0002D	MOVZWL	-2(STRING_BLOCK), ALLOC_LENGTH
		51	D7	00031	DECL	R1
51		07	8A	00033	BICB2	#7, R1

STR\$GET_FREE
1-012

G 11
16-Sep-1984 01:39:31 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:07 [LIBRTL.SRC]STRGETFRE.B32;1

Page 19
(6)

00	51	00000000G	0041	9E	00036	MOVAB	STR\$Q_SHORT Q[R1], INSQUE_ADDR	:
	B1		63	0E	0003E	INSQUE	(R3), 30(INSQUE_ADDR)	:
			1B	11	00042	BRB	5\$	:
		04	A2	9F	00044	4\$: PUSHAB	4(DESCRIP)	:
04	AE		62	3C	00047	MOVZWL	(DESCRIP), 4(SP)	:
		04	AE	9F	0004B	PUSHAB	4(SP)	:
00000000G	00		02	FB	0004E	CALLS	#2, LIB\$FREE VM	:
	07		50	E8	00055	BLBS	RETURN_STATUS, 5\$	:
	50	00000000G	8F	D0	00058	MOVL	#STR\$ FATINTERR, RETURN_STATUS	:
	54		50	D0	0005F	5\$: MOVL	RETURN_STATUS, RETURN_STATUS	:
	05		50	E9	00062	BLBC	RETURN_STATUS, 6\$	:
			62	B4	00065	CLRW	(DESCRIP)	1687
		04	A2	D4	00067	CLRL	4(DESCRIP)	1688
04		10	54	E8	0006A	6\$: BLBS	RETURN_STATUS, 8\$	1694
	54	03	00	ED	0006D	CMPZV	#0, #3, RETURN_STATUS, #4	:
			09	12	00072	BNEQ	8\$	:
			54	DD	00074	PUSHL	RETURN_STATUS	:
00000000G	00		01	FB	00076	7\$: CALLS	#1, LIB\$STOP	:
	50		01	D0	0007D	8\$: MOVL	#1, R0	1701
	5E		04	C0	00080	ADDL2	#4, SP	1702
			05	00	00083	RSB		:

; Routine Size: 132 bytes, Routine Base: _STR\$CODE + 0395

: 515 1703 1
: 516 1704 1 END
: 517 1705 1
: 518 1706 0 ELUDOM

! end of module STR\$GET_FREE

PSECT SUMMARY

Name	Bytes	Attributes
_STR\$CODE	1049	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	9 0	581	00:00.7

STR\$GET_FREE
1-012

H 11
16-Sep-1984 01:39:31
14-Sep-1984 12:40:07

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

Page 20
(6)

COMMAND QUALIFIERS

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

: Size: 1049 code + 0 data bytes
: Run Time: 00:15.0
: Elapsed Time: 00:59.8
: Lines/CPU Min: 6837
: Lexemes/CPU-Min: 36296
: Memory Used: 156 pages
: Compilation Complete

0214 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

