

(2)	50
(3)	86
(4)	136

DECLARATIONS
STR\$ANALYZE_SDESC - Analyze string descriptor
STR\$ANALYZE_SDESC_R1 - Analyze string descriptor

.....

```
0000 1 .TITLE STR$ANALYZE_SDESC - Analyze string descriptor
0000 2 .IDENT /1-004/ ; File: STRANASTR.MAR Edit: DG1004
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28
0000 29 :++
0000 30 : FACILITY: General Utility Library
0000 31
0000 32 : ABSTRACT:
0000 33
0000 34 : This module contains routines which extract the length and
0000 35 : address of the first byte of a string from any supported
0000 36 : class of string descriptor.
0000 37
0000 38 : ENVIRONMENT: Runs at any access mode, AST Reentrant
0000 39
0000 40 : AUTHOR: R. Reichert, CREATION DATE: 2-NOV-1981
0000 41
0000 42 : MODIFIED BY:
0000 43
0000 44 : 1-001 - Original. RKR 2-NOV-1981
0000 45 : 1-002 - Use general mode addressing. SBL 30-Nov-1981
0000 46 : 1-003 - Add support for class S0 string descriptors. DG 3-Oct-1983.
0000 47 : 1-004 - Change class S0 string descriptors to SB. DG 27-Feb-1984.
0000 48 :--
```

```
0000 50      .SBTTL  DECLARATIONS
0000 51      :
0000 52      : LIBRARY MACRO CALLS:
0000 53      :
0000 54      $DSCDEF      : DSC$_ symbols
0000 55      $$$DEF      : $$$_ symbols
0000 56      :
0000 57      : EXTERNAL DECLARATIONS:
0000 58      :
0000 59      : Prevent undeclared symbols from being automatically global.
0000 60      :
0000 61      .DSABL  GBL
0000 62      .EXTRN  STR$_ILLSTRCLA      ; Illegal string class
0000 63      .EXTRN  LIB$STOP           ; to signal fatal error
0000 64      :
0000 65      : MACROS:
0000 66      :
0000 67      : NONE
0000 68      :
0000 69      : EQUATED SYMBOLS:
0000 70      :
0000 71      : NONE
0000 72      :
0000 73      : OWN STORAGE:
0000 74      :
00000000 75      .PSECT _STR$DATA PIC,USR,CON,REL,LCL,NOSHR,-
0000 76      NOEXE, RD, WRT, LONG
0000 77      :
0000 78      : NONE
0000 79      :
0000 80      : PSECT DECLARATIONS:
0000 81      :
00000000 82      .PSECT _STR$CODE PIC,USR,CON,REL,LCL,SHR,-
0000 83      EXE, RD, NOWRT, LONG
0000 84      :
```

```

0000 86 .SBTTL STR$ANALYZE_SDESC - Analyze string descriptor
0000 87 :++
0000 88 : FUNCTIONAL DESCRIPTION:
0000 89 :
0000 90 : Extracts length and address of 1st data byte from any supported
0000 91 : class of string descriptor.
0000 92 :
0000 93 : CALLING SEQUENCE:
0000 94 :
0000 95 : STR$ANALYZE_SDESC (DESC.rt.dx, LENGTH.ww.r, ADDR.wa.r )
0000 96 :
0000 97 : FORMAL PARAMETERS:
0000 98 :
0000 99 : DESC.rt.dx address of a string descriptor
0000 100 :
0000 101 : LENGTH.ww.r address of a word to receive the strings length
0000 102 :
0000 103 : ADDR.wa.r address of a longword to receive the address
0000 104 : of the 1st data byte of the string.
0000 105 : IMPLICIT INPUTS:
0000 106 :
0000 107 : NONE
0000 108 :
0000 109 : IMPLICIT OUTPUTS:
0000 110 :
0000 111 : NONE
0000 112 :
0000 113 : COMPLETION STATUS:
0000 114 :
0000 115 : NONE
0000 116 :
0000 117 : SIDE EFFECTS:
0000 118 :
0000 119 : Signals STR$_ILLSTRCLA if invalid descriptor
0000 120 :
0000 121 :--
0000 122 :
0000 123 : Parameter displacements off AP
0000 124 : DESC = 4
0000 125 : LENGTH = 8
0000 126 : ADDR = 12
0000 127 :
0000 128 : .ENTRY STR$ANALYZE_SDESC, ^M<IV> ; Entry point
0000 129 : MOVL DESC(AP), R0 ; address of descriptor
0000 130 : JSB G^STR$ANALYZE_SDESC_R1 ; length ->R0
0000 131 : ; address->R1
0000 132 : MOVW R0, @LENGTH(AP) ; length to callers variable
0000 133 : MOVL R1, @ADDR(AP) ; address to callers variable
0000 134 : RET ; Return to caller

```

00000004
00000008
0000000C

4000
50 04 AC D0
00000015 GF 16
08 BC 50 B0
0C BC 51 D0
04 0014

```
0015 136 .SBTTL STR$ANALYZE_SDESC_R1 - Analyze string descriptor
0015 137 :++
0015 138 : FUNCTIONAL DESCRIPTION:
0015 139 :
0015 140 : Extracts length and address of 1st data byte from any supported
0015 141 : class of string descriptor.
0015 142 :
0015 143 : CALLING SEQUENCE:
0015 144 :
0015 145 : STR$ANALYZE_SDESC (DESC.rt.dx, LENGTH.wl.v, ADDR.wa.v )
0015 146 :
0015 147 : FORMAL PARAMETERS:
0015 148 :
0015 149 : DESC.rt.dx (Input in R0) address of a string descriptor
0015 150 :
0015 151 : LENGTH.wl.v (Returned in R0) the strings length
0015 152 :
0015 153 : ADDR.wa.v (Returned in R1) the address
0015 154 : of the 1st data byte of the string.
0015 155 : IMPLICIT INPUTS:
0015 156 :
0015 157 : NONE
0015 158 :
0015 159 : IMPLICIT OUTPUTS:
0015 160 :
0015 161 : NONE
0015 162 :
0015 163 : COMPLETION STATUS:
0015 164 :
0015 165 : NONE
0015 166 :
0015 167 : SIDE EFFECTS:
0015 168 :
0015 169 : Signals STR$_ILLSTRCLA if invalid string descriptor found
0015 170 :
0015 171 :--
```

```
0015 173 STR$ANALYZE_SDESC_R1::
0015 174
OF 51 04 A0 D0 0015 175      MOVL   DSC$A_POINTER(R0), R1      ; assume address of 1st byte
00 00 03 A0 8F 0019 176      CASEB  DSC$B_CLASS(R0), #DSC$K_CLASS_Z, #DSC$K_CLASS_SB
002D' 001E 177 10$:      .WORD  CLASS_Z-10$      ; 0 Z
002D' 0020 178      .WORD  CLASS_S-10$      ; 1 S
002D' 0022 179      .WORD  CLASS_D-10$      ; 2 D
002D' 0024 180      .WORD  CLASS_V-10$      ; 3 V (obsolete)
0031' 0026 181      .WORD  CLASS_A-10$      ; 4 A (obsolete)
002D' 0028 182      .WORD  CLASS_P-10$      ; 5 P (obsolete)
002D' 002A 183      .WORD  CLASS_PI-10$     ; 6 PI (obsolete)
002D' 002C 184      .WORD  CLASS_J-10$      ; 7 J (obsolete)
002D' 002E 185      .WORD  CLASS_JI-10$     ; 8 JI (obsolete)
002D' 0030 186      .WORD  CLASS_SD-10$     ; 9 SD
0031' 0032 187      .WORD  CLASS_NCA-10$    ; 10 NCA
003F' 0034 188      .WORD  CLASS_VS-10$     ; 11 VS
002D' 0036 189      .WORD  CLASS_VSA-10$    ; 12 VSA
002D' 0038 190      .WORD  CLASS_UBS-10$   ; 13 UBS
002D' 003A 191      .WORD  CLASS_UBA-10$   ; 14 UBA
002D' 003C 192      .WORD  CLASS_SB-10$    ; 15 SB
003E 193
003E 194 CLASS_V:      ; obsolete classes
003E 195 CLASS_P:
003E 196 CLASS_PI:
003E 197 CLASS_J:
003E 198 CLASS_JI:
003E 199 CLASS_VSA:   ; nonstring classes that fall inrange
003E 200 CLASS_UBS:
003E 201 CLASS_UBA:
00000000'8F DD 003E 202 ERROR:  PUSHL   #STR$_ILLSTRCLA      ; Illegal string class or
0044 203      ; invalid length in classes
0044 204      ; A or NCA
00000000'GF 01 FB 0044 205      CALLS   #1, G^LIB$STOP      ; signal fatal error - no return
004B 206
004B 207 CLASS_Z:      ; read like class _S
004B 208 CLASS_S:
004B 209 CLASS_D:
004B 210 CLASS_SD:
004B 211 CLASS_SB:
50 60 3C 004B 212      MOVZWL  DSC$W_LENGTH(R0), R0      ; length
05 05 004E 213      RSB      ; return to caller
004F 214
004F 215 CLASS_NCA:    ; assume its really contiguous
004F 216 CLASS_A:
50 50 0C A0 D0 004F 217      MOVL   DSC$L_ARSIZE(R0), R0      ; array size = length of string
50 FFFF0000 8F D3 0053 218      BITL   #*XFFF0000, R0      ; make sure < 2**16 -1
E2 12 005A 219      BNEQU  ERROR      ; else reject
05 05 005C 220      RSB      ; return to caller
005D 221
50 81 3C 005D 222 CLASS_VS:      ; varying string
0060 223      MOVZWL  (R1)+, R0      ; length -> R0, R1 -> addr of
0060 224      ; 1st data byte
05 0060 225      RSB      ; return to caller
0061 226
0061 227      .END      ; End of module STR$ANALYZE_SDESC
```



```

ADDR = 0000000C
CLASS_A 0000004F R 03
CLASS_D 0000004B R 03
CLASS_J 0000003E R 03
CLASS_JI 0000003E R 03
CLASS_NCA 0000004F R 03
CLASS_P 0000003E R 03
CLASS_PI 0000003E R 03
CLASS_S 0000004B R 03
CLASS_SB 0000004B R 03
CLASS_SD 0000004B R 03
CLASS_UBA 0000003E R 03
CLASS_UBS 0000003E R 03
CLASS_V 0000003E R 03
CLASS_VS 0000005D R 03
CLASS_VSA 0000003E R 03
CLASS_Z 0000004B R 03
DESC = 00000004
DSC$A_POINTER = 00000004
DSC$B_CLASS = 00000003
DSC$K_CLASS_SB = 0000000F
DSC$K_CLASS_Z = 00000000
DSC$L_ARSIZE = 0000000C
DSC$W_LENGTH = 00000000
ERROR 0000003E R 03
LENGTH = 00000008
LIB$STOP ***** X 00
STR$ANALYZE_SDESC 00000000 RG 03
STR$ANALYZE_SDESC_R1 00000015 RG 03
STR$_ILLSTRCLA ***** X 00
    
```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_STR\$DATA	00000000 (0.)	02 (2.)	PIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
_STR\$CODE	00000061 (97.)	03 (3.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.04	00:00:02.42
Command processing	111	00:00:00.30	00:00:03.76
Pass 1	209	00:00:03.25	00:00:15.52
Symbol table sort	0	00:00:00.57	00:00:02.66
Pass 2	52	00:00:00.65	00:00:03.49
Symbol table output	5	00:00:00.02	00:00:00.04
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	410	00:00:04.84	00:00:27.90

The working set limit was 1050 pages.
27116 bytes (53 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 562 non-local and 1 local symbols.
227 source lines were read in Pass 1, producing 15 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.

↑-----↑
! Macro library statistics !
↑-----↑

<u>Macro library name</u>	<u>Macros defined</u>
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5

604 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:STRANASTR/OBJ=OBJ\$:STRANASTR MSRC\$:STRANASTR/UPDATE=(ENH\$:STRANASTR)

The image displays a grid of 144 small, illegible document thumbnails arranged in 12 rows and 12 columns. The thumbnails are too small to read, but some contain faint text labels. The labels are as follows:

- Row 1: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 2: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 3: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 4: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 5: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 6: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 7: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 8: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 9: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 10: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 11: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS
- Row 12: Column 1: OTSPKDIU LIS; Column 2: OTSPKDIU LIS; Column 3: STRABMU LIS; Column 4: STRALOC LIS; Column 5: STRANASTR LIS; Column 6: STRAPPEND LIS; Column 7: STRARITH LIS; Column 8: STRCHESTA LIS; Column 9: STRCOMCAS LIS; Column 10: STRTERMIO LIS; Column 11: STRCOPY LIS; Column 12: STRTERMIO LIS