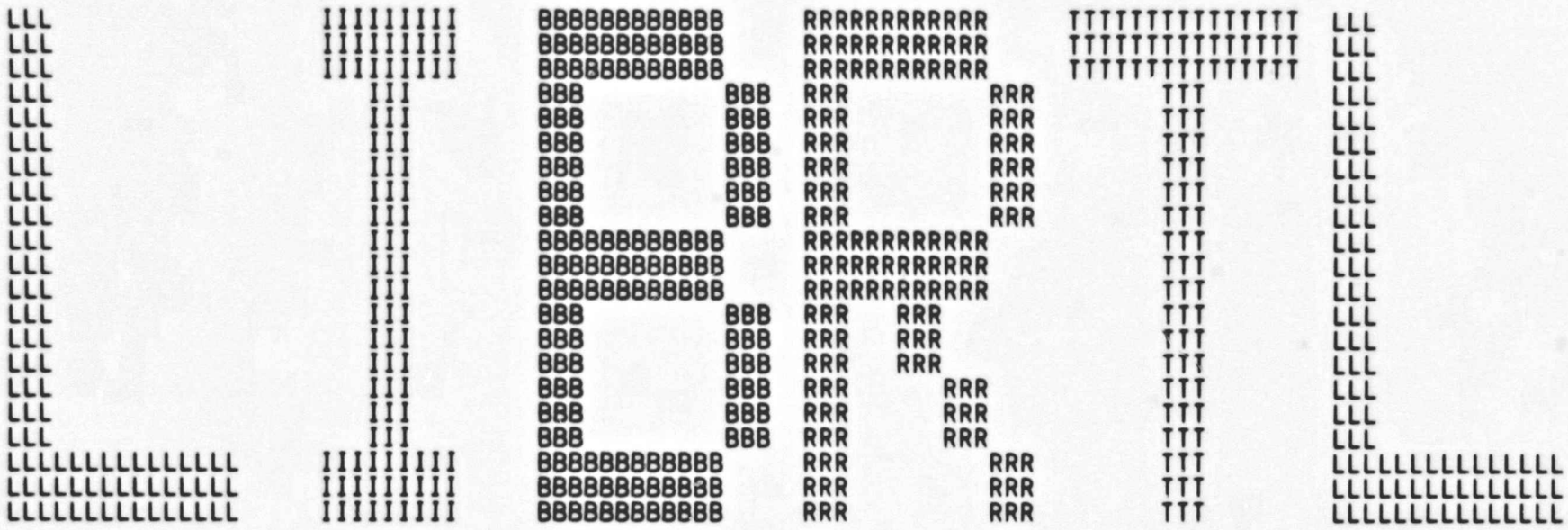
LLL IIIIIIIII BBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL IIIIIIIII BBBBBBBBBBBB RRRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL IIIIIIIII BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBBBBBBBBBB RRRRRRRRRRR TTT LLL
LLL III BBBBBBBBBBBB RRRRRRRRRRRR TTT LLL
LLL III BBBBBBBBBBBB RRRRRRRRRRRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL

```
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR     AAAAAA    NN      NN    AAAAAA     SSSSSSSS  TTTTTTTTTT  RRRRRRRR
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR     AAAAAA    NN      NN    AAAAAA     SSSSSSSS  TTTTTTTTTT  RRRRRRRR
SS              TT      RR      RR  AA    AA  NN      NN  AA    AA  SS              TT      RR      RR
SS              TT      RR      RR  AA    AA  NN      NN  AA    AA  SS              TT      RR      RR
SS              TT      RR      RR  AA    AA  NNNN    NN  AA    AA  SS              TT      RR      RR
SS              TT      RR      RR  AA    AA  NNNN    NN  AA    AA  SS              TT      RR      RR
  SSSSSS        TT      RRRRRRRR    AA    AA  NN  NN  NN  AA    AA    SSSSSS        TT      RRRRRRRR
  SSSSSS        TT      RRRRRRRR    AA    AA  NN  NN  NN  AA    AA    SSSSSS        TT      RRRRRRRR
        SS      TT      RR   RR     AAAAAAAAAA  NN    NNNN  AAAAAAAAAA          SS  TT      RR   RR
        SS      TT      RR   RR     AAAAAAAAAA  NN    NNNN  AAAAAAAAAA          SS  TT      RR   RR
        SS      TT      RR     RR   AA    AA  NN      NN  AA    AA            SS  TT      RR     RR
        SS      TT      RR     RR   AA    AA  NN      NN  AA    AA            SS  TT      RR     RR
SSSSSSSS        TT      RR      RR  AA    AA  NN      NN  AA    AA  SSSSSSSS  TT      RR      RR
SSSSSSSS        TT      RR      RR  AA    AA  NN      NN  AA    AA  SSSSSSSS  TT      RR      RR

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
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 ;--
```

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

```
                                        .SBTTL  STR$ANALYZE_SDESC - Analyze string descriptor
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       Extracts length and address of 1st data byte from any supported
                                ;       class of string descriptor.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       STR$ANALYZE_SDESC (DESC.rt.dx, LENGTH.ww.r, ADDR.wa.r )
                                ;
                                ; FORMAL PARAMETERS:
                                ;
                                ;       DESC.rt.dx      address of a string descriptor
                                ;
                                ;       LENGTH.ww.r     address of a word to receive the strings length
                                ;
                                ;       ADDR.wa.r       address of a longword to receive the address
                                ;                       of the 1st data byte of the string.
                                ; IMPLICIT INPUTS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE
                                ;
                                ; COMPLETION STATUS:
                                ;
                                ;       NONE
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       Signals STR$_ILLSTRCLA if invalid descriptor
                                ;
                                ;--
                                
                                ; Parameter displacements off AP
00000004                        DESC            = 4
00000008                        LENGTH          = 8
0000000C                        ADDR            = 12
                                
                                        .ENTRY STR$ANALYZE_SDESC, ^M<IV>        ; Entry point
50   04 AC    D0                MOVL    DESC(AP), R0                    ; address of descriptor
00000015'GF   16                JSB     G^STR$ANALYZE_SDESC_R1          ; length ->R0
                                                                        ; address->R1
08 BC   50    B0                MOVW    R0, @LENGTH(AP)                 ; length to callers variable
0C BC   51    D0                MOVL    R1, @ADDR(AP)                   ; address to callers variable
              04                RET                                     ; Return to caller
```

```
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$ANALYZE_SDESC_R1::
                                0015   174
          51    04 A0    D0     0015   175          MOVL    DSC$A_POINTER(R0), R1   ; assume address of 1st byte
       0F 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
                      0020'    0024   180          .WORD   CLASS_V-10$     ;  3    V  (obsolete)
                      0031'    0026   181          .WORD   CLASS_A-10$     ;  4    A
                      0020'    0028   182          .WORD   CLASS_P-10$     ;  5    P  (obsolete)
                      0020'    002A   183          .WORD   CLASS_PI-10$    ;  6    PI (obsolete)
                      0020'    002C   184          .WORD   CLASS_J-10$     ;  7    J  (obsolete)
                      0020'    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
                      0020'    0036   189          .WORD   CLASS_VSA-10$   ; 12    VSA
                      0020'    0038   190          .WORD   CLASS_UBS-10$   ; 13    UBS
                      0020'    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$W_LENGTH(R0), R0    ; length
                         05    004E   213          RSB                             ; return to caller
                               004F   214
                               004F   215 CLASS_NCA:                       ; assume its really contiguous
                               004F   216 CLASS_A:
       50    0C A0    D0     004F   217          MOVL    DSC$L_ARSIZE(R0), R0    ; array size = length of string
 50    FFFF0000 8F    D3     0053   218          BITL    #^XFFFF0000, R0         ; make sure < 2**16 -1
                   E2    12    005A   219          BNEQU   ERROR                   ; else reject
                         05    005C   220          RSB                             ; return to caller
                               005D   221
                               005D   222 CLASS_VS:                        ; varying string
             50    81    3C    005D   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
ABS	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 !
+----------------------------+
```

Macro library name	Macros defined
_255DUA28:[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)
```

OTSPKDIVL
LIS

OTSPKDIVS
LIS

STRABMUL
LIS

STRCHESTA
LIS

RTLLIB
LIS

STRAPPEND
LIS

STRARITH
LIS

STRALLOC
LIS

STRANASTR
LIS

STRCOMCAS
LIS

OTSSCOPY
LIS

OTSTERMIO
LIS