```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     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        BBBBBBBBBBBB     RRRRRRRRRRRR           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
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
```

```
LL              IIIIII    BBBBBBBB      AAAAAA    NN      NN    AAAAAA      SSSSSSSS  TTTTTTTTTT  RRRRRRRR
LL              IIIIII    BBBBBBBB      AAAAAA    NN      NN    AAAAAA      SSSSSSSS  TTTTTTTTTT  RRRRRRRR
LL                II      BB      BB    AA    AA  NN      NN   AA    AA   SS            TT      RR      RR
LL                II      BB      BB    AA    AA  NN      NN   AA    AA   SS            TT      RR      RR
LL                II      BB      BB    AA    AA  NNNN    NN   AA    AA   SS            TT      RR      RR
LL                II      BB      BB    AA    AA  NNNN    NN   AA    AA   SS            TT      RR      RR
LL                II      BBBBBBBB    AA      AA  NN  NN  NN   AA    AA     SSSSSS      TT      RRRRRRRR
LL                II      BBBBBBBB    AA      AA  NN  NN  NN   AA    AA     SSSSSS      TT      RRRRRRRR
LL                II      BB      BB  AAAAAAAAAA  NN    NNNN   AAAAAAAAAA         SS    TT      RR   RR
LL                II      BB      BB  AAAAAAAAAA  NN    NNNN   AAAAAAAAAA         SS    TT      RR   RR
LL                II      BB      BB  AA      AA  NN      NN   AA      AA         SS    TT      RR     RR      ....
LL                II      BB      BB  AA      AA  NN      NN   AA      AA         SS    TT      RR     RR      ....
LLLLLLLLLL      IIIIII    BBBBBBBB    AA      AA  NN      NN   AA      AA   SSSSSSSS    TT      RR       RR     ....
LLLLLLLLLL      IIIIII    BBBBBBBB    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  LIB$ANALYZE_SDESC - Analyze string descriptor
0000     2         .IDENT  /1-003/         ; File: LIBANASTR.MAR Edit: DG1003
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 - Add support for class SO string descriptors.  DG 3-Oct-1983.
0000    46 ; 1-003 - Change class SO string descriptors to SB.  DG 27-Feb-1984.
0000    47 ;--
```

```
                     .SBTTL  DECLARATIONS
                ;
                ; LIBRARY MACRO CALLS:
                ;
                ;       $DSCDEF                 ; DSC$_ symbols
                ;       $SSDEF                  ; SS$_ symbols
                ;
                ; EXTERNAL DECLARATIONS:
                ;
                ; Prevent undeclared symbols from being automatically global.
                ;
                        .DSABL  GBL
                        .EXTRN  LIB$_INVSTRDES          ; Invalid string descriptor
                ;
                ; MACROS:
                ;
                ;       NONE
                ;
                ; EQUATED SYMBOLS:
                ;
                ;       NONE
                ;
                ; OWN STORAGE:
                ;
00000000        .PSECT _LIB$DATA PIC, USR, CON, REL, LCL, NOSHR, -
                                NOEXE, RD, WRT, LONG
                ;
                ;       NONE
                ;
                ; PSECT DECLARATIONS:
                ;
                        .PSECT _LIB$CODE PIC, USR, CON, REL, LCL, SHR, -
00000000                        EXE, RD, NOWRT, LONG
```

```
                        84          .SBTTL  LIB$ANALYZE_SDESC - Analyze string descriptor
                        85 ;++
                        86 ; FUNCTIONAL DESCRIPTION:
                        87 ;
                        88 ;       Extracts length and address of 1st data byte from any supported
                        89 ;       class of string descriptor.
                        90 ;
                        91 ; CALLING SEQUENCE:
                        92 ;
                        93 ;       ret_status.wlc.v = LIB$ANALYZE_SDESC (DESC.rt.dx,
                        94 ;                                             LENGTH.ww.r,
                        95 ;                                             ADDR.wa.r )
                        96 ;
                        97 ; FORMAL PARAMETERS:
                        98 ;
                        99 ;       DESC.rt.dx      address of a string descriptor
                       100 ;
                       101 ;       LENGTH.ww.r     address of a word to receive the strings length
                       102 ;
                       103 ;       ADDR.wa.r       address of a longword to receive the address
                       104 ;                       of the 1st data byte of the string.
                       105 ; IMPLICIT INPUTS:
                       106 ;
                       107 ;       NONE
                       108 ;
                       109 ; IMPLICIT OUTPUTS:
                       110 ;
                       111 ;       NONE
                       112 ;
                       113 ; COMPLETION STATUS:
                       114 ;
                       115 ;       SS$_NORMAL      Normal successful completion
                       116 ;       LIB$_INVSTRDES  Invalid string descriptor
                       117 ;
                       118 ; SIDE EFFECTS:
                       119 ;
                       120 ;       NONE
                       121 ;
                       122 ;--
                       123
                       124 ; Parameter displacements off AP
            00000004   125 DESC            = 4
            00000008   126 LENGTH          = 8
            0000000C   127 ADDR            = 12
                       128
                  4004 0000  129         .ENTRY LIB$ANALYZE_SDESC, ^M<IV,R2>     ; Entry point
     50   04 AC    D0 0002  130         MOVL    DESC(AP), R0                    ; address of descriptor
00000015'GF    16 0006  131         JSB     G^LIB$ANALYZE_SDESC_R2  ; length ->R1
                       000C  132                                                 ; address->R2
                       000C  133                                                 ; status ->R0
     08 BC    51    B0 000C  134         MOVW    R1, @LENGTH(AP)                 ; length to callers variable
     0C BC    52    D0 0010  135         MOVL    R2, @ADDR(AP)                   ; address to callers variable
                    04 0014  136         RET                                     ; Return, with status in R0.
```

```
0015   138          .SBTTL  LIB$ANALYZE_SDESC_R2 - Analyze string descriptor
0015   139 ;++
0015   140 ; FUNCTIONAL DESCRIPTION:
0015   141 ;
0015   142 ;       Extracts length and address of 1st data byte from any supported
0015   143 ;       class of string descriptor.
0015   144 ;
0015   145 ; CALLING SEQUENCE:
0015   146 ;
0015   147 ;       LIB$ANALYZE_SDESC (DESC.rt.dx, LENGTH.wl.v, ADDR.wa.v )
0015   148 ;
0015   149 ; FORMAL PARAMETERS:
0015   150 ;
0015   151 ;       DESC.rt.dx      (In R0) address of a string descriptor
0015   152 ;
0015   153 ;       LENGTH.wl.v     (Returned in R1) the strings length
0015   154 ;
0015   155 ;       ADDR.wa.v       (Returned in R2) the address
0015   156 ;                       of the 1st data byte of the string.
0015   157 ; IMPLICIT INPUTS:
0015   158 ;
0015   159 ;       NONE
0015   160 ;
0015   161 ; IMPLICIT OUTPUTS:
0015   162 ;
0015   163 ;       NONE
0015   164 ;
0015   165 ; COMPLETION STATUS:
0015   166 ;
0015   167 ;       SS$_NORMAL      Normal successful completion
0015   168 ;       LIB$_INVSTRDES  Invalid string descriptor
0015   169 ;
0015   170 ; SIDE EFFECTS:
0015   171 ;
0015   172 ;       NONE
0015   173 ;
0015   174 ;--
```

```
                        0015   176 LIB$ANALYZE_SDESC_R2::
                        0015   177
          52   04 A0    D0  0015   178          MOVL    DSC$A_POINTER(R0), R2   ; assume address of 1st byte
       0F 00   03 A0    8F  0019   179          CASEB   DSC$B_CLASS(R0), #DSC$K_CLASS_Z, #DSC$K_CLASS_SB
                     0028' 001E   180 10$:      .WORD   CLASS_Z-10$     ;  0    Z
                     0028' 0020   181           .WORD   CLASS_S-10$     ;  1    S
                     0028' 0022   182           .WORD   CLASS_D-10$     ;  2    D
                     0020' 0024   183           .WORD   CLASS_V-10$     ;  3    V  (obsolete)
                     002F' 0026   184           .WORD   CLASS_A-10$     ;  4    A
                     0020' 0028   185           .WORD   CLASS_P-10$     ;  5    P  (obsolete)
                     0020' 002A   186           .WORD   CLASS_PI-10$    ;  6    PI (obsolete)
                     0020' 002C   187           .WORD   CLASS_J-10$     ;  7    J  (obsolete)
                     0020' 002E   188           .WORD   CLASS_JI-10$    ;  8    JI (obsolete)
                     0028' 0030   189           .WORD   CLASS_SD-10$    ;  9    SD
                     002F' 0032   190           .WORD   CLASS_NCA-10$   ; 10    NCA
                     0040' 0034   191           .WORD   CLASS_VS-10$    ; 11    VS
                     0020' 0036   192           .WORD   CLASS_VSA-10$   ; 12    VSA
                     0020' 0038   193           .WORD   CLASS_UBS-10$   ; 13    UBS
                     0020' 003A   194           .WORD   CLASS_UBA-10$   ; 14    UBA
                     0028' 003C   195           .WORD   CLASS_SB-10$    ; 15    SB
                        003E   196
                        003E   197 CLASS_V:                            ; obsolete classes
                        003E   198 CLASS_P:
                        003E   199 CLASS_PI:
                        003E   200 CLASS_J:
                        003E   201 CLASS_JI:
                        003E   202 CLASS_VSA:                          ; nonstring classes that fall inrange
                        003E   203 CLASS_UBS:
                        003E   204 CLASS_UBA:
    50  00000000'8F     D0  003E   205 ERROR:  MOVL    #LIB$_INVSTRDES, R0     ; unsupported descriptor or
                        0045   206                                     ; invalid length in classes
                        0045   207                                     ; _A or _NCA
                     05 0045   208          RSB                        ; return to caller
                        0046   209
                        0046   210 CLASS_Z:                           ; read like class _S
                        0046   211 CLASS_S:
                        0046   212 CLASS_D:
                        0046   213 CLASS_SD:
                        0046   214 CLASS_SB:
          51   60    3C  0046   215          MOVZWL  DSC$W_LENGTH(R0), R1    ; length
          50   01    D0  0049   216          MOVL    #SS$_NORMAL, R0         ; success status
                     05 004C   217          RSB                        ; return to caller
                        004D   218
                        004D   219 CLASS_NCA:                          ; assume its really contiguous
                        004D   220 CLASS_A:
        51   0C A0    D0  004D   221          MOVL    DSC$L_ARSIZE(R0), R1
51  FFFF0000 8F     D3  0051   222          BITL    #^XFFFF0000, R1         ; make sure < 2**16 -1
                 E4  12  0058   223          BNEQU   ERROR                   ; else reject
          50   01    D0  005A   224          MOVL    #SS$_NORMAL, R0         ; success status
                     05 005D   225          RSB                        ; return to caller
                        005E   226
                        005E   227 CLASS_VS:                           ; varying string
          51   82    3C  005E   228          MOVZWL  (R2)+, R1               ; length -> R1, R2 -> addr of
                        0061   229                                     ; 1st data byte
          50   01    D0  0061   230          MOVL    #SS$_NORMAL, R0         ; success status
                     05 0064   231          RSB                        ; return to caller
                        0065   232
```

```
0065   233          .END                    ; End of module LIB$ANALYZE_SDESC
```

```
ADDR                    = 0000000C
CLASS_A                   0000004D R      03
CLASS_D                   00000046 R      03
CLASS_J                   0000003E R      03
CLASS_JI                  0000003E R      03
CLASS_NCA                 0000004D R      03
CLASS_P                   0000003E R      03
CLASS_PI                  0000003E R      03
CLASS_S                   00000046 R      03
CLASS_SB                  00000046 R      03
CLASS_SD                  00000046 R      03
CLASS_UBA                 0000003E R      03
CLASS_UBS                 0000003E R      03
CLASS_V                   0000003E R      03
CLASS_VS                  0000005E R      03
CLASS_VSA                 0000003E R      03
CLASS_Z                   00000046 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$ANALYZE_SDESC         00000000 RG     03
LIB$ANALYZE_SDESC_R2      00000015 RG     03
LIB$_INVSTRDES            ******** X      00
SS$_NORMAL              = 00000001
```

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
_LIB$DATA	00000000 (0.)	02 (2.)	PIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
_LIB$CODE	00000065 (101.)	03 (3.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.06	00:00:00.85
Command processing	123	00:00:00.30	00:00:02.14
Pass 1	205	00:00:03.44	00:00:14.90
Symbol table sort	0	00:00:00.55	00:00:01.90
Pass 2	54	00:00:00.67	00:00:03.45
Symbol table output	4	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	427	00:00:05.07	00:00:23.29

The working set limit was 1200 pages.
27230 bytes (54 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 561 non-local and 1 local symbols.
233 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$:LIBANASTR/OBJ=OBJ$:LIBANASTR MSRC$:LIBANASTR/UPDATE=(ENH$:LIBANASTR)
```

RTLMACMAR
MAR

LIBASCEBC
LIS

LIBASTINP
LIS

RTLMACB32
REQ

STRMACROS
REQ

LIBBINTRE
LIS

RTLODBG
REQ

LIBDEF
FOR

LIBANASTR
LIS

LIBASNMBX
LIS

LIBBBCCI
LIS

LIBCHAR
LIS

LIBABUPCA
LIS

LIBADDX
LIS

RTLPSECT
REQ

SIGDEF
FOR

LIBASCTIM
LIS

LIBTABMAC
MAR

LIBATTACH
LIS

STRLNK
REQ

LIBA2EREV
LIS

LIBBBSSI
LIS

LIBCALLG
LIS

LIBCLICAL
LIS

LIBADDP
LIS