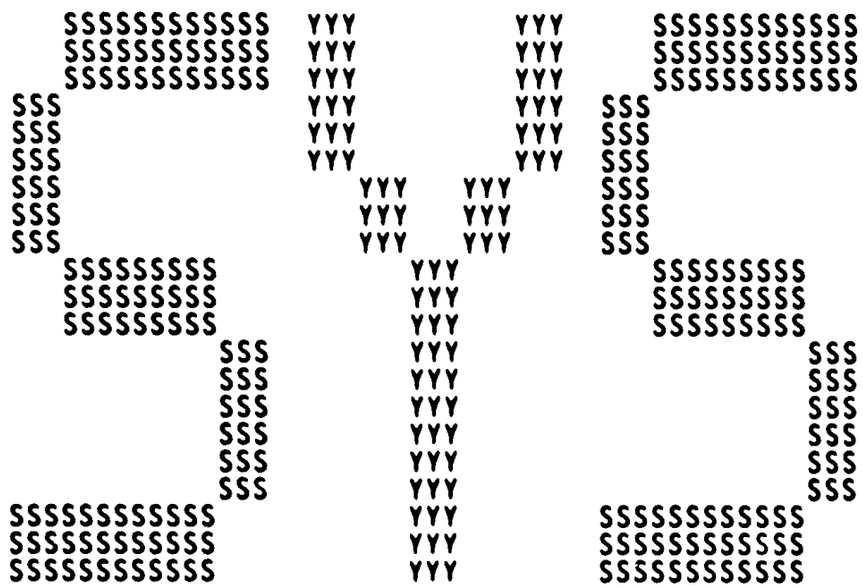

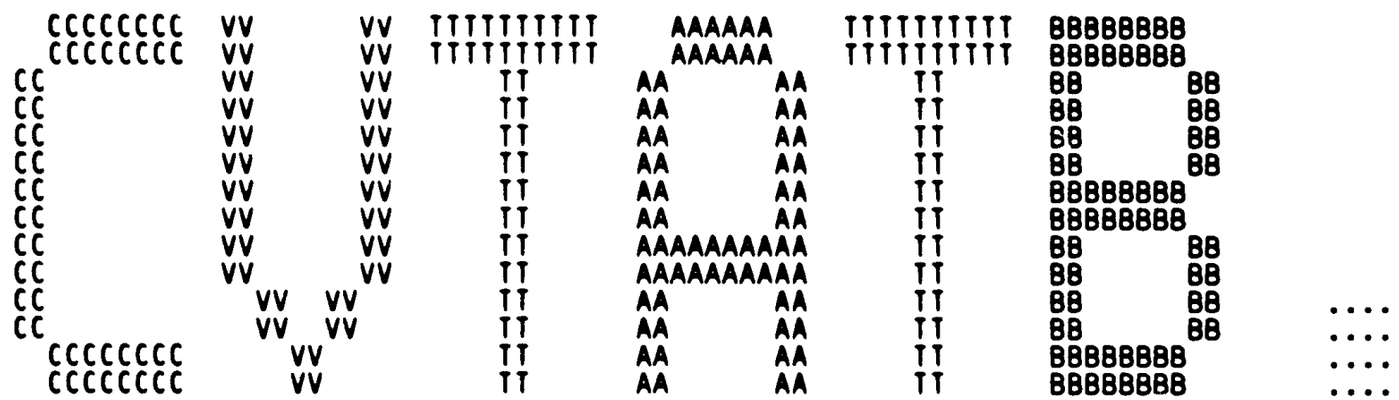
CCCCCCCC VV VV TTTTTTTTTT AAAAAA TTTTTTTTTT BBBBBBBB
CCCCCCCC VV VV TTTTTTTTTT AAAAAA TTTTTTTTTT BBBBBBBB
CC VV VV TT AA AA TT BB BB
CC VV VV TT AA AA TT BB BB
CC VV VV TT AA AA TT BB BB
CC VV VV TT AA AA TT BB BB
CC VV VV TT AA AA TT BBBBBBBB
CC VV VV TT AA AA TT BBBBBBBB
CC VV VV TT AAAAAAAAAA TT BB BB
CC VV VV TT AAAAAAAAAA TT BB BB
CC VV VV TT AA AA TT BB BB ....
CC VV VV TT AA AA TT BB BB ....
CCCCCCCC VV TT AA AA TT BBBBBBBB ....
CCCCCCCC VV TT AA AA TT BBBBBBBB ....

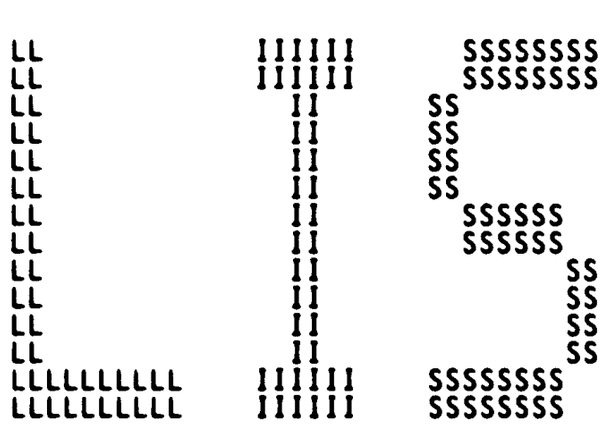
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

Table of contents

```
                        1          .TITLE  CVT_ATB - ASCII NUMBER TO BINARY CONVERSION
                        2          .IDENT  'V04-000'
                        3
                        4  ;
                        5  ;*****************************************************************************
                        6  ;*                                                                           *
                        7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
                        8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
                        9  ;*  ALL RIGHTS RESERVED.                                                     *
                       10  ;*                                                                           *
                       11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
                       12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
                       13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
                       14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
                       15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
                       16  ;*  TRANSFERRED.                                                              *
                       17  ;*                                                                           *
                       18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
                       19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
                       20  ;*  CORPORATION.                                                              *
                       21  ;*                                                                           *
                       22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
                       23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
                       24  ;*                                                                           *
                       25  ;*                                                                           *
                       26  ;*****************************************************************************
                       27
                       28  ;
                       29  ; EQUATED SYMBOLS:
                       30  ;
00000004               31          COUNT   = 4                     ; COUNT ARG IN ARG LIST
00000008               32          STRING  = 8                     ; STRING ARG
0000000C               33          RESULT  = 12                    ; RESULT ADDRESS
                       34
                       35  ;++
                       36  ;
                       37  ; FACILITY: SYSTEM LIBRARY
                       38  ;
                       39  ; ABSTRACT:
                       40  ;
                       41  ;       THIS ROUTINE PERFORMS ASCII INTEGER TO BINARY CONVERSION IN
                       42  ;       DECIMAL, OCTAL, AND HEX RADIX; RADIX DETERMINED BY ENTRY POINT.
                       43  ;       AN OPTIONAL LEADING SIGN IS ACCEPTED.
                       44  ;
                       45  ; ENVIRONMENT:
                       46  ;
                       47  ;       STAR NATIVE MODE PROCESSOR, ANY ACCESS LEVEL. NO SPECIAL
                       48  ;       INSTRUCTIONS OR SYSTEM SERVICES ARE USED. 9 LONGWORDS OF
                       49  ;       STACK SPACE NEEDED.
                       50  ;
                       51          .SBTTL  HISTORY                 ; DETAILED
                       52  ;
                       53  ; AUTHOR:  ANDREW C. GOLDSTEIN  26-JAN-78  16:48
                       54  ;
                       55  ; MODIFIED BY:
                       56  ;
                       57  ;       V03-001 LJK0254         Lawrence J. Kenah       6-Dec-1983
```

```
0000     58 ;                  Change names of entry points from LIB$CVT_xTB to FIL$CVT_xTB
0000     59 ;                  to avoid conflict with same names in RTL. Use .ENTRY directive
0000     60 ;                  for register save mask.
0000     61 ;--
```

```
                     63          .SBTTL  DECLARATIONS
                     64 ;++
                     65 ;
                     66 ; FUNCTIONAL DESCRIPTION:
                     67 ;
                     68 ; THE ROUTINE WORKS IN THE OBVIOUS MANNER OF ALL CONVERSION ROUTINES:
                     69 ; IT SCANS THE INPUT STRING AND CONVERTS EACH CHARACTER INTO ITS
                     70 ; NUMERICAL EQUIVALENT AND CHECKS IT FOR LEGALITY AGAINST THE RADIX.
                     71 ; THE NUMBER BEING ACCUMULATED IS THEN MULTIPLIED BY THE RADIX AND THE
                     72 ; NEW DIGIT IS ADDED IN. NON-RADIX CHARACTERS IN THE INPUT STRING CAUSE
                     73 ; AN ERROR RETURN. A SIGN IN OTHER THAN THE FIRST CHARACTER POSITION
                     74 ; AND OVERFLOW FROM 32 BITS (UNSIGNED) ALSO CAUSE AN ERROR RETURN.
                     75 ;
                     76 ; NOTE THAT THESE ROUTINES ARE IDENTICAL TO ROUTINES WITH THE SAME NAMES AND
                     77 ; THE LIB$ PREFIX THAT EXIST IN THE RUN-TIME LIBRARY. THESE ROUTINES USE
                     78 ; DIFFERENT PSECT NAMES AND DIFFERENT ENTRY POINT NAMES TO ALLOW THEM TO
                     79 ; COEXIST IN IMAGES THAT LINK AGAINST BOTH STARLET.OLB AND SYS.STB.
                     80 ;
                     81 ; CALLING SEQUENCE:
                     82 ;       CALL FIL$CVT_DTB (COUNT,STRING,RESULT)  ; CONVERT DECIMAL TO BINARY
                     83 ;       CALL FIL$CVT_OTB (COUNT,STRING,RESULT)  ; CONVERT OCTAL TO BINARY
                     84 ;       CALL FIL$CVT_HTB (COUNT,STRING,RESULT)  ; CONVERT HEX TO BINARY
                     85 ;
                     86 ;
                     87 ; INPUT PARAMETERS:
                     88 ;       COUNT:  BYTE COUNT OF INPUT STRING
                     89 ;       STRING: ADDRESS OF INPUT STRING
                     90 ;
                     91 ; IMPLICIT INPUTS:
                     92 ;       NONE
                     93 ;
                     94 ; OUTPUT PARAMETERS:
                     95 ;       RESULT: ADDRESS TO STORE LONGWORD RESULT
                     96 ;
                     97 ; IMPLICIT OUTPUTS:
                     98 ;       NONE
                     99 ;
                    100 ; COMPLETION CODES:
                    101 ;       R0 = 1  SUCCESFUL CONVERSION
                    102 ;          = 0  ILLEGAL CHARACTER OR ZERO BYTE COUNT
                    103 ;
                    104 ; SIDE EFFECTS:
                    105 ;       NONE
                    106 ;
                    107 ;--
                    108
                    109         .ENABL  LSB
                    110
           00000000 111         .PSECT  YCVTATB,WRT
                    112 ;
                    113 ; BASE AND VALUE TABLES TO CONVERT DIGITS INTO THEIR NUMERICAL VALUES
                    114 ;
30 41 61    0000    115 BASETAB:        .BYTE   ^A'a',    ^A'A',    ^A'0'
39 5A 7A    0003    116 TOPTAB:         .BYTE   ^A'z',    ^A'Z',    ^A'9'
30 37 57    0006    117 VALTAB:         .BYTE   ^A'a'-10, ^A'A'-10, ^A'0'-00
                    118
                    119
```

```
                        0009   120          .ENTRY  -                       ; ENTRY FOR DECIMAL CONVERT
                        0009   121                  FIL$CVT_DTB,-
                  003C  0009   122                  ^M<R2,R3,R4,R5>         ; SAVE REGISTERS 2-5
        52  0A     D0    000B   123          MOVL    #10,R2                  ; R2 = RADIX
            0C     11    000E   124          BRB     10$
                        0010   125
                        0010   126          .ENTRY  -                       ; ENTRY FOR OCTAL CONVERT
                  003C  0010   127                  FIL$CVT_OTB,-
                        0012   128                  ^M<R2,R3,R4,R5>         ; SAVE REGISTERS 2-5
        52  08     D0    0012   129          MOVL    #8,R2                   ; R2 = RADIX
            05     11    0015   130          BRB     10$
                        0017   131
                        0017   132          .ENTRY  -                       ; ENTRY FOR HEXADECIMAL CONVERT
                  003C  0017   133                  FIL$CVT_HTB,-
                        0019   134                  ^M<R2,R3,R4,R5>         ; SAVE REGISTERS 2-5
        52  10     D0    0019   135          MOVL    #16,R2                  ; R2 = RADIX
                        001C   136
            50     7C    001C   137  10$:    CLRQ    R0                      ; ZERO INITIAL NUMBER
            53     7C    001E   138          CLRQ    R3                      ; AND STRING POINTER & SIGN
            53     D7    0020   139          DECL    R3                      ; BACK OFF COUNT
            40     11    0022   140          BRB     70$                     ; AND ENTER PRE-TESTED LOOP
                        0024   141
    55  08 BC43    9A    0024   142  20$:    MOVZBL  @STRING(AP)[R3],R5      ; GET NEXT CHARACTER
            53     D5    0029   143          TSTL    R3                      ; SEE IF THIS IS FIRST CHARACTER
            0E     12    002B   144          BNEQ    30$                     ; SKIP SIGN CHECK IF NOT
        2B  55     91    002D   145          CMPB    R5,#^A'+'               ; CHECK FOR PLUS
            32     13    0030   146          BEQL    70$
        2D  55     91    0032   147          CMPB    R5,#^A'-'               ; CHECK FOR MINUS
            04     12    0035   148          BNEQ    30$                     ; BRANCH IF NOT
            54     D6    0037   149          INCL    R4                      ; SET NEGATIVE FLAG
            29     11    0039   150          BRB     70$                     ; OK - LOOP FOR NEXT CHARACTER
                        003B   151
            51     D4    003B   152  30$:    CLRL    R1                      ; INIT TABLE INDEX
     BE AF41   55     91    003D   153  40$:    CMPB    R5,BASETAB[R1]          ; CHECK DIGIT AGAINST BASE CHARACTER
               07     1F    0042   154          BLSSU   50$                     ; BRANCH IF DIGIT IS LESS
     BA AF41   55     91    0044   155          CMPB    R5,TOPTAB[R1]           ; CHECK DIGIT AGAINST TOP CHARACTER
               06     1B    0049   156          BLEQU   60$                     ; BRANCH IF LESS - VALID
       EE 51   03     F2    004B   157  50$:    AOBLSS  #3,R1,40$               ; NOT THIS RANGE - TRY NEXT LOWER BASE
               26     11    004F   158          BRB     90$                     ; NOT A LEGAL DIGIT
                        0051   159
    55  B1 AF41   82    0051   160  60$:    SUBB    VALTAB[R1],R5           ; COMPUTE NUMERIC VALUE (NOTE - ALWAYS +)
        52  55     D1    0056   161          CMPL    R5,R2                   ; CHECK DIGIT AGAINST RADIX
            1C     1E    0059   162          BGEQU   90$                     ; TOUGH LUCK .....
50  55  50  52     7A    005B   163          EMUL    R2,R0,R5,R0             ; SCALE VALUE BY RADIX AND ADD DIGIT
            51     D5    0060   164          TSTL    R1                      ; CHECK FOR OVERFLOW
            13     12    0062   165          BNEQ    90$
                        0064   166
 BB 53  04 AC     F2    0064   167  70$:    AOBLSS  COUNT(AP),R3,20$        ; LOOP FOR DIGITS
                        0069   168
          03 54  E9    0069   169          BLBC    R4,80$                  ; CHECK SIGN BIT
        50  50     CE    006C   170          MNEGL   R0,R0                   ; MAKE NEGATIVE IF DESIRED
     0C BC  50     D0    006F   171  80$:    MOVL    R0,@RESULT(AP)          ; STORE RESULT
        50  01     D0    0073   172          MOVL    #1,R0                   ; RETURN TRUE
                  04    0076   173          RET
                        0077   174 ;
                        0077   175 ; TO HERE ON ANY BAD CHARACTER
                        0077   176 ;
```

```
                50  D4  0077    177 90$:    CLRL    R0                  ; RETURN FALSE
                    04  0079    178         RET
                        007A    179
                        007A    180         .DSABL  LSB
                        007A    181
                        007A    182
                        007A    183
                        007A    184         .END
```

```
BASETAB           00000000 R      02
COUNT           = 00000004
FIL$CVT_DTB       00000009 RG     02
FIL$CVT_HTB       00000017 RG     02
FIL$CVT_OTB       00000010 RG     02
OP$_ACBD        = 0000006F
OP$_ACBF        = 0000004F
OP$_ACBG        = 00004FFD
OP$_ACBH        = 00006FFD
OP$_ADDD2       = 00000060
OP$_ADDD3       = 00000061
OP$_ADDF2       = 00000040
OP$_ADDF3       = 00000041
OP$_ADDG2       = 000040FD
OP$_ADDG3       = 000041FD
OP$_ADDH2       = 000060FD
OP$_ADDH3       = 000061FD
OP$_ADDP4       = 00000020
OP$_ADDP6       = 00000021
OP$_ASHP        = 000000F8
OP$_CLRD        = 0000007C
OP$_CLRF        = 000000D4
OP$_CLRG        = 0000007C
OP$_CLRH        = 00007CFD
OP$_CMPD        = 00000071
OP$_CMPF        = 00000051
OP$_CMPG        = 000051FD
OP$_CMPH        = 000071FD
OP$_CMPP3       = 00000035
OP$_CMPP4       = 00000037
OP$_CRC         = 0000000B
OP$_CVTBD       = 0000006C
OP$_CVTBF       = 0000004C
OP$_CVTBG       = 00004CFD
OP$_CVTBH       = 00006CFD
OP$_CVTDB       = 00000068
OP$_CVTDF       = 00000076
OP$_CVTDH       = 000032FD
OP$_CVTDL       = 0000006A
OP$_CVTDW       = 00000069
OP$_CVTFB       = 00000048
OP$_CVTFD       = 00000056
OP$_CVTFG       = 000099FD
OP$_CVTFH       = 000098FD
OP$_CVTFL       = 0000004A
OP$_CVTFW       = 00000049
OP$_CVTGB       = 000048FD
OP$_CVTGF       = 000033FD
OP$_CVTGH       = 000056FD
OP$_CVTGL       = 00004AFD
OP$_CVTGW       = 000049FD
OP$_CVTHB       = 000068FD
OP$_CVTHD       = 0000F7FD
OP$_CVTHF       = 0000F6FD
OP$_CVTHG       = 000076FD
OP$_CVTHL       = 00006AFD
OP$_CVTHW       = 000069FD
OP$_CVTLD       = 0000006E
OP$_CVTLF       = 0000004E
OP$_CVTLG       = 00004EFD
OP$_CVTLH       = 00006EFD
OP$_CVTLP       = 000000F9
OP$_CVTPL       = 00000036
OP$_CVTPS       = 00000008
OP$_CVTPT       = 00000024
OP$_CVTRDL      = 0000006B
OP$_CVTRFL      = 0000004B
OP$_CVTRGL      = 00004BFD
OP$_CVTRHL      = 00006BFD
OP$_CVTSP       = 00000C09
OP$_CVTTP       = 00000026
OP$_CVTWD       = 0000006D
OP$_CVTWF       = 0000004D
OP$_CVTWG       = 00004DFD
OP$_CVTWH       = 00006DFD
OP$_DIVD2       = 00000066
OP$_DIVD3       = 00000067
OP$_DIVF2       = 00000046
OP$_DIVF3       = 00000047
OP$_DIVG2       = 000046FD
OP$_DIVG3       = 000047FD
OP$_DIVH2       = 000066FD
OP$_DIVH3       = 000067FD
OP$_DIVP        = 00000027
OP$_EDITPC      = 00000038
OP$_EMODD       = 00000074
OP$_EMODF       = 00000054
OP$_EMODG       = 000054FD
OP$_EMODH       = 000074FD
OP$_MATCHC      = 00000039
OP$_MNEGD       = 00000072
OP$_MNEGF       = 00000052
OP$_MNEGG       = 000052FD
OP$_MNEGH       = 000072FD
OP$_MOVD        = 00000070
OP$_MOVF        = 00000050
OP$_MOVG        = 000050FD
OP$_MOVH        = 000070FD
OP$_MOVP        = 00000034
OP$_MOVTC       = 0000002E
OP$_MOVTUC      = 0000002F
OP$_MULD2       = 00000064
OP$_MULD3       = 00000065
OP$_MULF2       = 00000044
OP$_MULF3       = 00000045
OP$_MULG2       = 000044FD
OP$_MULG3       = 000045FD
OP$_MULH2       = 000064FD
OP$_MULH3       = 000065FD
OP$_MULP        = 00000025
OP$_POLYD       = 00000075
OP$_POLYF       = 00000055
OP$_POLYG       = 000055FD
OP$_POLYH       = 000075FD
OP$_SCANC       = 0000002A
OP$_SKPC        = 0000003B
OP$_SPANC       = 0000002B
OP$_SUBD2       = 00000062
OP$_SUBD3       = 00000063
OP$_SUBF2       = 00000042
OP$_SUBF3       = 00000043
OP$_SUBG2       = 000042FD
OP$_SUBG3       = 000043FD
OP$_SUBH2       = 000062FD
OP$_SUBH3       = 000063FD
OP$_SUBP4       = 00000022
OP$_SUBP6       = 00000023
OP$_TSTD        = 00000073
OP$_TSTF        = 00000053
OP$_TSTG        = 000053FD
OP$_TSTH        = 000073FD
RESULT          = 0000000C
STRING          = 00000008
TOPTAB            00000003 R      02
VALTAB            00000006 R      02
```

**

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
YCVTATB	0000007A (122.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.06	00:00:01.45
Command processing	132	00:00:00.65	00:00:04.93
Pass 1	347	00:00:08.22	00:00:26.24
Symbol table sort	0	00:00:00.50	00:00:01.49
Pass 2	49	00:00:03.00	00:00:10.13
Symbol table output	11	00:00:00.10	00:00:00.30
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	578	00:00:12.55	00:00:44.56

The working set limit was 1500 pages.
36813 bytes (72 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 370 non-local and 9 local symbols.
2936 source lines were read in Pass 1, producing 21 object records in Pass 2.
134 pages of virtual memory were used to define 133 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	4

420 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:CVTATB/OBJ=OBJ$:CVTATB MASD$:[EMULAT.SRC]MISSING/UPDATE=(MASD$:[EMULAT.ENH]MISSING)+MASD$:[SYS.SRC]CVTATB/UPDATE=(MAS
```

0373 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CMODSSDSP
LIS

BUGCHECK
LIS

COMDRVSUB
LIS

CLUSTRVEC
LIS

BUFFERCTL
LIS

DEADLOCK
LIS

CVTFILNAM
LIS

BOOPARAM
LIS

CJFSYSVEC
LIS

CVTATB
LIS

BUGCHKMSG
LIS

CONSOLIO
LIS