```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR      RRR            TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR      RRR            TTT        LLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
```

```
BBBBBBBB      AAAAAA      SSSSSSSS  XX      XX  LL            AAAAAA    TTTTTTTTTT  EEEEEEEEEE
BBBBBBBB      AAAAAA      SSSSSSSS  XX      XX  LL            AAAAAA    TTTTTTTTTT  EEEEEEEEEE
BB      BB  AA      AA  SS          XX      XX  LL          AA      AA      TT      EE
BB      BB  AA      AA  SS          XX      XX  LL          AA      AA      TT      EE
BB      BB  AA      AA  SS            XX  XX    LL          AA      AA      TT      EE
BB      BB  AA      AA  SS            XX  XX    LL          AA      AA      TT      EE
BBBBBBBB    AA      AA    SSSSSS        XX      LL          AA      AA      TT      EEEEEEEE
BBBBBBBB    AA      AA    SSSSSS        XX      LL          AA      AA      TT      EEEEEEEE
BB      BB  AAAAAAAAAA          SS    XX  XX    LL          AAAAAAAAAA      TT      EE
BB      BB  AAAAAAAAAA          SS    XX  XX    LL          AAAAAAAAAA      TT      EE
BB      BB  AA      AA          SS  XX      XX  LL          AA      AA      TT      EE              ....
BB      BB  AA      AA          SS  XX      XX  LL          AA      AA      TT      EE              ....
BBBBBBBB    AA      AA  SSSSSSSS    XX      XX  LLLLLLLLLL  AA      AA      TT      EEEEEEEEEE      ....
BBBBBBBB    AA      AA  SSSSSSSS    XX      XX  LLLLLLLLLL  AA      AA      TT      EEEEEEEEEE      ....

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  BAS$XLATE
0000     2          .IDENT  /1-004/                 ; File: BASXLATE.MAR EDIT: RNH1004
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: BASIC code support
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;
0000    34  ;       This module implements the BASIC-PLUS-2 XLATE function
0000    35  ;
0000    36  ; ENVIRONMENT: User Mode, AST Reentrant
0000    37  ;
0000    38  ;--
0000    39  ; AUTHOR: R. WILL, CREATION DATE: 18-May-79
0000    40  ;
0000    41  ; MODIFIED BY:
0000    42  ;
0000    43  ; R. Will, : VERSION 1
0000    44  ; 1-001 - Original
0000    45  ; 1-002 - Change calls to STR$COPY.  JBS 16-JUL-1979
0000    46  ; 1-003 - Change a INCW R1 to INCL R1.  R1 contains an address. FM 5-FEB-81
0000    47  ; 1-004 - Change shared external references to G^ RNH 25-Sep-81
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; INCLUDE FILES:
                ;

                        $DSCDEF                                 ; define descriptor offsets

                ;
                ; EXTERNAL DECLARATIONS:
                ;
                        .DSABL  GBL                             ; Prevent undeclared
                                                                ; symbols from being
                                                                ; automatically global.

                        .EXTRN  STR$COPY_DX_R8                  ; copy input string to temp
                                                                ; and temp string to output
                        .EXTRN  STR$COPY_R_R8                   ; copy temp str to dest str
                        .EXTRN  STR$GET1_DX                     ; allocate temp string
                        .EXTRN  STR$FREE1_DX                    ; deallocate temp string
                        .EXTRN  LIB$GET_VM                      ; allocate heap memory
                        .EXTRN  LIB$FREE_VM                     ; deallocate heap memory

                ;
                ; MACROS:
                ;

                ;
                ; EQUATED SYMBOLS:
                ;

00000000        00      =       null = ^X00

                ;
                ; OWN STORAGE:
                ;

                ;
                ; PSECT DECLARATIONS:
                ;
00000000                        .PSECT _BAS$CODE PIC, USR, CON, REL, LCL, SHR, -
                                        EXE, RD, NOWRT, LONG
```

```
                     0000     92          .SBTTL  BAS$XLATE         - Perform BASIC XLATE function
                     0000     93 ;++
                     0000     94 ; FUNCTIONAL DESCRIPTION:
                     0000     95 ;
                     0000     96 ;       This routine implements the BASIC-PLUS-2 XLATE function.
                     0000     97 ;       For AST re-entrancy, the routine will create a local dynamic string
                     0000     98 ;       descriptor and call STR$COPY to copy the source string to the local
                     0000     99 ;       (instead of using any mechanism to prevent AST level routines
                     0000    100 ;       from writing to the source string and moving it from under us).
                     0000    101 ;       The routine will also create a local dynamic string descriptor and
                     0000    102 ;       call allocate to get a string to translate into.  The routine will
                     0000    103 ;       then use both local strings (which will not need to get larger) to do
                     0000    104 ;       the translating.  The routine will use the MOVTUC to translate until
                     0000    105 ;       the translated character is the NULL character.  The NULL will not be
                     0000    106 ;       written to the destination string, and the translation will continue
                     0000    107 ;       with the next character.  After the translating is finished,
                     0000    108 ;       the routine will call STR$COPY to copy the edited string to the
                     0000    109 ;       destination string.
                     0000    110 ;
                     0000    111 ; CALLING SEQUENCE:
                     0000    112 ;
                     0000    113 ;       CALL BAS$XLATE (dest_string.wx.dx, src_string.rx.dx, table.rx.dx)
                     0000    114 ;
                     0000    115 ; INPUT PARAMETERS:
                     0000    116 ;
                00000008 0000    117 ;       src_string = 8
                0000000C 0000    118 ;       table = 12
                     0000    119 ;
                     0000    120 ; IMPLICIT INPUTS:
                     0000    121 ;
                     0000    122 ;       NONE
                     0000    123 ;
                     0000    124 ; OUTPUT PARAMETERS:
                     0000    125 ;
                00000004 0000    126 ;       dest_string = 4
                     0000    127 ;
                     0000    128 ; IMPLICIT OUTPUTS:
                     0000    129 ;
                     0000    130 ;       NONE
                     0000    131 ;
                     0000    132 ; FUNCTION VALUE:
                     0000    133 ; COMPLETION CODES:
                     0000    134 ;
                     0000    135 ;       NONE
                     0000    136 ;
                     0000    137 ; SIDE EFFECTS:
                     0000    138 ;
                     0000    139 ;       This routine calls STR$COPY and STR$FREE1 and therefore will
                     0000    140 ;       allocate dynamic string space to a temporary, may allocate dynamic
                     0000    141 ;       string space to the destination string, and may cause any of the
                     0000    142 ;       their error messages to be signalled.  This routine also calls
                     0000    143 ;       LIB$GET_VM and LIB$FREE_VM and so any of their errors may be
                     0000    144 ;       signalled.
                     0000    145 ;
                     0000    146 ;--
                     0000    147
                   41FC  0000    148          .ENTRY BAS$XLATE, ^M<R2,R3,R4,R5,R6,R7,R8,IV>
```

```
                                   0002
                                   0002  ;+
                                   0002  ; Create a local descriptor and copy the input string to it using STR$COPY
                                   0002  ;-
                                   0002
              51    08 AC     D0   0002          MOVL    src_string(AP), R1              ; pointer to src string
                       7E     D4   0006          CLRL    -(SP)                           ; address of local string
            020E0000 8F       DD   0008          PUSHL   #<<DSC$K_CLASS_D @ 24> ! <DSC$K_DTYPE_T @ 16>>
                                   000E                                                  ; fill type, class and length
              50       5E     D0   000E          MOVL    SP, R0                          ; R0 points to local descriptr
            00000000'GF       16   0011          JSB     G^STR$COPY_DX_R8                ; copy string to local
                                   0017
                                   0017  ;+
                                   0017  ; Create a local descriptor and allocate space to it, to use as destination
                                   0017  ; string for MOVTUC
                                   0017  ;-
                                   0017
                       7E     D4   0017          CLRL    -(SP)                           ; address of local string
            020E0000 8F       DD   0019          PUSHL   #<<DSC$K_CLASS_D @ 24> ! <DSC$K_DTYPE_T @ 16>>
                                   001F                                                  ; fill type, class and len
                       5E     DD   001F          PUSHL   SP                              ; point to descriptor
                    08 BC     3F   0021          PUSHAW  @src_string(AP)                 ; length to allocate
         00000000'GF    02    FB   0024          CALLS   #2, G^STR$GET1_DX               ; allocate space
                                   002B
                                   002B  ;+
                                   002B  ; Call LIB$GET_VM to allocate 256 bytes to use for translate table
                                   002B  ; and create the translation table
                                   002B  ;-
                                   002B
                       7E     D4   002B          CLRL    -(SP)                           ; space for memory pointer
        7E  00000100 8F       D0   002D          MOVL    #256, -(SP)                     ; # bytes to allocate
                    04 AE     DF   0034          PUSHAL  4(SP)                           ; ptr to output parameter
                    04 AE     DF   0037          PUSHAL  4(SP)                           ; ptr to byte count
         00000000'GF    02    FB   003A          CALLS   #2, G^LIB$GET_VM                ; allocate the space
              50    0C BC     7D   0041          MOVQ    @table(AP), R0                  ; get table pointer and length
04 BE   0100 8F   00   61  50  2C  0045          MOVC5   R0, (R1), #null, #256, @4(SP)   ; fill the translate table
                                   004E
                                   004E  ;+
                                   004E  ; fill registers for the MOVTUC loop
                                   004E  ;       R0      src len
                                   004E  ;       R1      src pointer
                                   004E  ;       R3      address of translation table
                                   004E  ;       R4      dest len
                                   004E  ;       R5      dest pointer
                                   004E  ;-
                                   004E
              50    10 AE     7D   004E          MOVQ    16(SP), R0                      ; R0 & R1 <- len & ptr for src
              54    08 AE     7D   0052          MOVQ    8(SP), R4                       ; R4&R5 <- len & ptr for dest
              53    04 AE     D0   0056          MOVL    4(SP), R3                       ; R3 has addr of extendd table
                                   005A
                                   005A  ;+
                                   005A  ; Registers are initialized, so MOVTUC until get a NULL, increment src ptr
                                   005A  ; decrement src len to describe string remaining after NULL translation.
                                   005A  ; Then continue translating.
                                   005A  ;-
                                   005A
  65   54   63   00   61   50  2F  005A  1$:     MOVTUC  R0, (R1), #null, (R3), R4, (R5) ; find null translation
```

```
                   50  B5  0061          TSTW    R0
                   06  13  0063          BEQLU   FINISH                          ; all was translated
                   50  B7  0065          DECW    R0                              ; subtract 1 from remain len
                   51  D6  0067          INCL    R1                              ; add 1 to remaining pointer
                   EF  11  0069          BRB     1$                              ; process remaining
                           006B
                           006B  ;+
                           006B  ; The string has been translated.  Free the VM used for the translate table.
                           006B  ; Copy the temporory storage to the destination string.  (Note that the trans-
                           006B  ; lated length is the source length minus the number of unfilled bytes in the
                           006B  ; temporary string left in R4 by the MOVTUC.)  Deallocate the temporary string
                           006B  ; and the copied source string. Clean up the stack and return.
                           006B  ;-
                           006B
                           006B  FINISH:
                04 AE  DF  006B          PUSHAL  4(SP)                           ; point to VM pointer
                04 AE  DF  006E          PUSHAL  4(SP)                           ; point to VM length
       00000000'GF  02  FB  0071          CALLS   #2, G^LIB$FREE_VM               ; free the VM table copy
                   8E  7C  0078          CLRQ    (SP)+                           ; clean descr from stack
                           007A
   51    08 BC  54  A3  007A          SUBW3   R4, @src_string(AP), R1         ; compute & store temp len
          52    04 AE  D0  007F          MOVL    4(SP), R2                       ; point to src for copy
          50    04 AC  D0  0083          MOVL    dest_string(AP), R0             ; dest for copy
            00000000'GF  16  0087          JSB     G^STR$COPY_R_R8                 ; copy
                           008D
                   6E  DF  008D          PUSHAL  (SP)                            ; point to temp dest descr
       00000000'GF  01  FB  008F          CALLS   #1, G^STR$FREE1_DX              ; deallocate it
                   8E  7C  0096          CLRQ    (SP)+                           ; clean the descr from stack
                           0098
                   6E  DF  0098          PUSHAL  (SP)                            ; point to temp src descr
       00000000'GF  01  FB  009A          CALLS   #1, G^STR$FREE1_DX              ; deallocate it
                   8E  7C  00A1          CLRQ    (SP)+                           ; clean the descr from stack
                           00A3
                       04  00A3          RET
                           00A4
                           00A4          .END                                    ; End of BAS$XLATE
```

```
BAS$XLATE                       00000000 RG    02
DEST_STRING                   = 00000004
DSC$K_CLASS_D                 = 00000002
DSC$K_DTYPE_T                 = 0000000E
FINISH                          0000006B R     02
LIB$FREE_VM                     ********   X   00
LIB$GET_VM                      ********   X   00
NULL                          = 00000000
SRC_STRING                    = 00000008
STR$COPY_DX_R8                  ********   X   00
STR$COPY_R_R8                   ********   X   00
STR$FREET_DX                    ********   X   00
STR$GET1_DX                     ********   X   00
TABLE                         = 0000000C
```

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
_BAS$CODE	000000A4 (164.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:00.65
Command processing	117	00:00:00.43	00:00:02.18
Pass 1	137	00:00:01.84	00:00:04.30
Symbol table sort	0	00:00:00.17	00:00:00.31
Pass 2	56	00:00:00.61	00:00:01.58
Symbol table output	3	00:00:00.02	00:00:00.02
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	348	00:00:03.17	00:00:09.06

The working set limit was 1050 pages.
9144 bytes (18 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 143 non-local and 1 local symbols.
241 source lines were read in Pass 1, producing 13 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	4

190 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

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

BASXLATE
LIS

BPAMESAG
LIS

BPAPSS
LIS

BPAGETBLK
LIS

BASZIRET
LIS