



```

000000      TTTTTTTTTT      SSSSSSSS      CCCCCCCC      VV      VV      TTTTTTTTTT      TTTTTTTTTT      LL      LL
000000      TTTTTTTTTT      SSSSSSSS      CCCCCCCC      VV      VV      TTTTTTTTTT      TTTTTTTTTT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
00      00      TT      SS      CC      VV      VV      TT      TT      LL      LL
000000      TT      SSSSSSSS      CCCCCCCC      VV      VV      TT      TT      LL      LL
000000      TT      SSSSSSSS      CCCCCCCC      VV      VV      TT      TT      LLLLLLLLLL      LLLLLLLLLL
                                         .....
                                         .....
                                         .....
                                         .....

```

```

LL      IIIIIII      SSSSSSSS
LL      IIIIIII      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      IIIIIII      SSSSSSSS
LLLLLLLLLL      IIIIIII      SSSSSSSS

```

(2) 50  
(3) 82

DECLARATIONS  
OTSSCVT\_TL\_L - Convert text (logical) to longword

```

0000 1      .TITLE OTSSCVTTLL - Convert text (logical) to longword
0000 2      .IDENT /1-003/ ; File: OTSCVTLL.MAR Edit: SBL1003
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: Language independent support library
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : A routine to convert a text string indicating "true" or "false" to
0000 35 : a integer representation.
0000 36 :
0000 37 : ENVIRONMENT: User Mode, AST Reentrant
0000 38 :
0000 39 :--
0000 40 : AUTHOR: Steven B. Lionel, CREATION DATE: 01-Mar-1979
0000 41 :
0000 42 : MODIFIED BY:
0000 43 :
0000 44 : Edit History
0000 45 :
0000 46 : 1-001 - Original. Complete rewrite of FOR$CNV IN L. SBL 01-Mar-1979
0000 47 : 1-002 - SBL24781 Fix bug where SP is used instead of AP. SBL 2-Jul-79
0000 48 : 1-003 - Do correct thing if value_size is wrong. SBL 25-Feb-1980

```

```
0000 50      .SBTTL  DECLARATIONS
0000 51      :
0000 52      : INCLUDE FILES:
0000 53      :
0000 54      :
0000 55      :
0000 56      : EXTERNAL DECLARATIONS:
0000 57      :
0000 58      .DSABL  GBL                ; Prevent undeclared
0000 59      :                               ; symbols from being
0000 60      :                               ; automatically global.
0000 61      .EXTRN  OTSS_INPCONERR    ; Error code
0000 62      :
0000 63      :
0000 64      : MACROS:
0000 65      :
0000 66      :
0000 67      :
0000 68      : EQUATED SYMBOLS:
0000 69      :
0000 70      :
0000 71      :
0000 72      : OWN STORAGE:
0000 73      :
0000 74      :
0000 75      :
0000 76      : PSECT DECLARATIONS:
0000 77      :
00000000 78      .PSECT _OTSS$CODE PIC,  USR,  CON,  REL,  LCL,  SHR,  -
0000 79      EXE,  RD,  NOWRT,  LONG
0000 80
```

```

0000 82      .SBTTL OTSSCVT_TL_L - Convert text (logical) to longword
0000 83      :
0000 84      :++ FUNCTIONAL DESCRIPTION:
0000 85      :
0000 86      : This routine converts a text string to a longword value
0000 87      : using FORTRAN-77 type L format conversion. The format of
0000 88      : the text accepted is as follows:
0000 89      :
0000 90      :     < 0 or more blanks >
0000 91      :     <     < end of string >
0000 92      :     or
0000 93      :     <     < "" or nothing >
0000 94      :     Letter: < "T", "t", "F", "f" >
0000 95      :     < 0 or more of any character >
0000 96      :     < end of string > > >
0000 97      : The value returned by OTSSCVT_TL_L is:
0000 98      :
0000 99      :     All 1 bits (-1) if the character
0000 100     :     denoted "Letter" above is "T" or "t".
0000 101     :     Zero otherwise.
0000 102     :
0000 103     : If the input string does not conform to the above specification,
0000 104     : value is set to zero (false), and the condition code
0000 105     : OTSS_INPCONERR is returned.
0000 106     :
0000 107     : Note: ANSI X3.9-1978 FORTRAN-77 does not allow for the input
0000 108     : field to be all blank, but since VAX-11 FORTRAN IV-PLUS did
0000 109     : allow it, (returning .FALSE.), it is allowed here too.
0000 110     :
0000 111     : Note: With this implementation, the strings ".TRUE." and
0000 112     : ".FALSE." are valid. In the previous version, they were not.
0000 113     :
0000 114     : Note: For compatibility with previous releases, the global
0000 115     : symbol FOR$CNV_IN_L may be used to call OTSSCVT_TL_L.
0000 116     :
0000 117     :
0000 118     : CALLING SEQUENCE:
0000 119     :
0000 120     :     status.wlc.v = OTSSCVT_TL_L (input_string.rt.ds, value.wl.r
0000 121     :     [, value_size.rl.v])
0000 122     :
0000 123     : INPUT PARAMETERS:
0000 124     :
0000 125     :     input_string = 4      : Input text string by descriptor
0000 126     :     value_size  = 12     : Size of value in bytes. If not
0000 127     :     :                   : specified, 4 is assumed. Valid
0000 128     :     :                   : values are 1, 2 and 4. If invalid,
0000 129     :     :                   : an error is returned.
0000 130     :     :                   : If size is greater than 4, 4 is assumed.
0000 131     :
0000 132     : IMPLICIT INPUTS:
0000 133     :
0000 134     :     NONE
0000 135     :
0000 136     : OUTPUT PARAMETERS:
0000 137     :
0000 138     :     value = 8      : Output value by reference

```

00000004  
0000000C

00000008

```

0000 139 :
0000 140 : IMPLICIT OUTPUTS:
0000 141 :
0000 142 :     NONE
0000 143 :
0000 144 : COMPLETION CODES:
0000 145 :
0000 146 :     SSS_NORMAL      - Successful completion
0000 147 :     OTSS_INPCONERR - Input conversion error
0000 148 :
0000 149 : SIDE EFFECTS:
0000 150 :
0000 151 :     NONE
0000 152 :
0000 153 : --
0000 154 :
0000 155 FOR$CNV_IN L::      ; For compatibility
0004 0000 .ENTRY  OTSSCVT_TL_L, ^M<R2>
0002 157
61 50 04 BC 7D 0002 158      CLRL      R2                ; Initial value is .FALSE.
61 50 20 3B 0004 159      MOVQ     @input_string(AP), R0 ; Get descriptor
61 61 34 13 0008 160      SKPC    #^A/ /, R0, (R1)      ; Skip blanks
61 61 2E 91 000C 161      BEQL    FALSE                ; All blank, .FALSE.
61 61 06 12 000E 162      CMPB   #^A/./, (R1)          ; A period?
61 50 50 D7 0011 163      BNEQ   10$                  ; No
61 61 1A 13 0013 164      DECL   R0                    ; Yes, decrement character count
61 61 51 D6 0015 165      BEQL   ERROR                ; Last character, error
61 61 8F 91 0017 166      INCL   R1                    ; Skip over period
61 54 8F 91 0019 167 10$:  CMPB   #^A/T/, (R1)          ; Test for .TRUE.
61 74 1B 13 001D 168      BEQL   TRUE
61 74 8F 91 001F 169      CMPB   #^A/t/, (R1)
61 46 15 13 0023 170      BEQL   TRUE
61 46 8F 91 0025 171      CMPB   #^A/F/, (R1)          ; Test for .FALSE.
61 66 17 13 0029 172      BEQL   FALSE
61 66 8F 91 002B 173      CMPB   #^A/f/, (R1)
0002F 174      BEQL   FALSE
0031 175
0031 176 :+
0031 177 : Invalid character, return error OTSS_INPCONERR
0031 178 :-
50 00000000'8F D0 0031 179 ERROR:  MOVL   #OTSS_INPCONERR, R0 ; Error status code
000D 11 0038 180      BRB    EXIT                ; Return with value .FALSE.
003A 181
003A 182 :+
003A 183 : Text string represents .TRUE.
003A 184 :-
52 01 CE 003A 185 TRUE:   MNEGL  #1, R2                ; Value is .TRUE.
50 01 D0 003D 186      MOVL   #1, R0                ; SSS_NORMAL
0005 11 0040 187      BRB    EXIT                ; Exit
0042 188
0042 189 :+
0042 190 : Text string represents .FALSE.
0042 191 : R2 is already cleared.
0042 192 :-
50 01 D0 0042 193 FALSE:  MOVL   #1, R0                ; SSS_NORMAL
0000 11 0045 194      BRB    EXIT                ; Exit
0047 195

```

```
0047 196 :+
0047 197 : Return to caller with status value.
0047 198 :-
03 6C 91 0047 199 EXIT:  CMPB  (AP), #<value_size/4>  : Is size present?
1E 19 004A 200  BLSS  40$  : No, longword
04 0C AC D1 004C 201  CMPL  value_size(AP), #4  : Is it a longword?
18 18 0050 202  BGEQ  40$  : Yes, at least
02 0C AC D1 0052 203  CMPL  value_size(AP), #2  : Is it a word?
0C 0C 13 0056 204  BEQL  20$  : Yes
01 0C AC D1 0058 205  CMPL  value_size(AP), #1  : Is it a byte?
D3 12 005C 206  BNEQ  ERROR  : If not, it's an error
08 BC 52 90 005E 207  MOVB  R2, @value(AP)  : Convert byte
0A 11 0062 208  BRB  50$  : Exit
08 BC 52 B0 0064 209 20$: MOVW  R2, @value(AP)  : Move a word
04 11 0068 210  BRB  50$  : Exit
08 BC 52 D0 006A 211 40$: MOVL  R2, @value(AP)  : Move a longword
04 04 006E 212 50$: RET  : Return to caller
006F 213
006F 214 .END
```



```

ERROR          00000031 R    01
EXIT           00000047 R    01
FALSE         00000042 R    01
FOR$CNV IN L  00000000 RG   01
INPUT STRING  = 00000004
OTSSCVT TL L  00000000 RG   01
OTSS_INPCORERR ***** X  00
TRUE          0000003A R    01
VALUE        = 00000008
VALUE_SIZE   = 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
_OTSSCODE	0000006F ( 111.)	01 ( 1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

-----  
 ! Performance indicators !  
 -----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.05	00:00:02.10
Command processing	115	00:00:00.31	00:00:03.38
Pass 1	71	00:00:00.31	00:00:02.08
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	52	00:00:00.29	00:00:00.89
Symbol table output	2	00:00:00.01	00:00:00.01
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	274	00:00:00.98	00:00:08.47

The working set limit was 750 pages.  
 2694 bytes (6 pages) of virtual memory were used to buffer the intermediate code.  
 There were 10 pages of symbol table space allocated to hold 10 non-local and 4 local symbols.  
 214 source lines were read in Pass 1, producing 11 object records in Pass 2.  
 0 pages of virtual memory were used to define 0 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

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

