

CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLL

```

CCCCCCCC NN      NN  VV      VV  CCCCCCCC LL      IIIIII  AAAAAA  TTTTTTTTTT 88888888
CCCCCCCC NN      NN  VV      VV  CCCCCCCC LL      IIIIII  AAAAAA  TTTTTTTTTT 88888888
CC        NN      NN  VV      VV  CC        LL      II      AA      AA  TT      88      88
CC        NN      NN  VV      VV  CC        LL      II      AA      AA  TT      88      88
CC        NNNN    NN  VV      VV  CC        LL      II      AA      AA  TT      88      88
CC        NNNN    NN  VV      VV  CC        LL      II      AA      AA  TT      88888888 88
CC        NN  NN  NN  VV      VV  CC        LL      II      AA      AA  TT      88888888 88
CC        NN  NN  NN  VV      VV  CC        LL      II      AA      AA  TT      88      88
CC        NN      NNNN VV      VV  CC        LL      II      AAAAAAAAAA TT      88      88
CC        NN      NNNN VV      VV  CC        LL      II      AAAAAAAAAA TT      88      88
CC        NN      NN  VV  VV  VV  CC        LL      II      AA      AA  TT      88      88
CC        NN      NN  VV  VV  VV  CC        LL      II      AA      AA  TT      88      88
CCCCCCCC NN      NN  VV      VV  CCCCCCCC LLLLLLLLLL IIIIII  AA      AA  TT      88888888 .....
CCCCCCCC NN      NN  VV      VV  CCCCCCCC LLLLLLLLLL IIIIII  AA      AA  TT      88888888 .....

```

```

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

```

(1) 49
(1) 66

DECLARATIONS
CNVCLIATB - CONVERT ASCII TO SINGLE-PRECISION BINARY

```
0000 1 .TITLE CNVCLIATB - CONVERT ASCII TO BINARY
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 * ALL RIGHTS RESERVED.
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 * TRANSFERRED.
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 * CORPORATION.
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *****
0000 27
0000 28 ++
0000 29 FACILITY: COMMAND LANGUAGE INTERPRETER UTILITIES
0000 30
0000 31 ABSTRACT:
0000 32
0000 33 THIS SUBROUTINE CONVERTS AN UNSIGNED ASCII NUMERIC STRING TO
0000 34 AN UNSIGNED, SINGLE PRECISION VALUE.
0000 35
0000 36 ENVIRONMENT: USER MODE
0000 37
0000 38 AUTHOR: C. A. MONIA , CREATION DATE: 15-AUG-1977
0000 39
0000 40 MODIFIED BY:
0000 41
0000 42 : : VERSION
0000 43 01 -
0000 44
0000 45 101 T. Halvorsen 27-Jun-1979
0000 46 Change PSECT name to avoid conflicting psects warning with RTL
0000 47 --
```

```
0000 49 .SBTTL DECLARATIONS
0000 50 :
0000 51 : INCLUDE FILES:
0000 52 :
0000 53 :
0000 54 $CLIMSGDEF ; DEFINE CLI STATUS VALUES
0000 55 :
0000 56 :
0000 57 : OWN STORAGE:
0000 58 :
0000 59 : ASCII TO BINARY CONVERSION TABLE
0000 60 :
0000 61 :
00000000 62 .PSECT CNVCLIATB,EXE,RD,NOWRT
0000 63 :
42 41 39 38 37 36 35 34 33 32 31 30 0000 64 CVTABL: .ASCII /0123456789ABCDEF/ ; ALL LEGAL NUMERICS
46 45 44 43 000C
```

```

0010 66          .SBTTL  CNVCLIATB - CONVERT ASCII TO SINGLE-PRECISION BINARY
0010 67
0010 68      :++
0010 69      : FUNCTIONAL DESCRIPTION:
0010 70
0010 71      : THIS SUBROUTINE CONVERTS AN ASCII STRING TO A SINGLE PRECISION
0010 72      : BINARY VALUE. THE STRING IS CONSIDERED AS AN UNSIGNED VALUE.
0010 73
0010 74      : THE CONVERSION RADIX IS ONE OF THE FOLLOWING:
0010 75
0010 76          LIB$CVTCLIDTB - CONVERT DECIMAL TO BINARY
0010 77          LIB$CVTCLIOTB - CONVERT OCTAL TO BINARY
0010 78          LIB$CVTCLINTB - CONVERT HEXADECIMAL TO BINARY
0010 79
0010 80      : CALLING SEQUENCE:
0010 81
0010 82          JSB      LIB$CVTCLIXXX
0010 83
0010 84      : INPUT PARAMETERS:
0010 85
0010 86          R2,R3 = DESCRIPTOR FOR STRING TO BE CONVERTED
0010 87
0010 88      : IMPLICIT INPUTS:
0010 89
0010 90          NONE
0010 91
0010 92      : OUTPUT PARAMETERS:
0010 93
0010 94          R0 = CLIS_NORMAL, SUCCESS:
0010 95
0010 96          R1 CONTAINS THE DOUBLE PRECISION RESULT
0010 97
0010 98          R0 = CLIS_IVCHAR, INVALID NUMERIC CHARACTER FOUND
0010 99          R0 = CLIS_IVVALU, VALUE EXCEEDS A SINGLE PRECISION INTEGER
0010 100
0010 101      : IMPLICIT OUTPUTS:
0010 102
0010 103          R3 = ADDRESS OF BYTE FOLLOWING LAST CHARACTER OR ONE BYTE
0010 104          PAST AN INVALID CHARACTER
0010 105
0010 106      : COMPLETION CODES:
0010 107
0010 108          SEE ABOVE
0010 109
0010 110      : SIDE EFFECTS:
0010 111
0010 112          R2 IS DESTROYED
0010 113
0010 114      :--
0010 115
0010 116          .ENABL  LSB
0010 117
0010 118 LIB$CVTCLIDTB::
50  0A  D0 0010 119          MOVL  #10,R0          : GET RADIX
0010 120          BRB    10$
50  08  D0 0015 121 LIB$CVTCLIOTB::
0010 122          MOVL  #8,R0          : GET RADIX
  
```

```

03 11 0018 123 BRB 10$
50 10 D0 001A 124 LIB$CVTCLINTB::
0071 8F BB 001A 125 MOVL #16,R0
55 D4 001D 126 10$:
0023 127 PUSHR #*M<R0,R4,R5,R6>
54 00030001 8F D0 0021 128 CLRL R5
52 B7 0023 129 20$:
2D 1F 002A 130 MOVL #CLIS_NORMAL,R4
54 00038050 8F D0 002C 131 DECW R2
50 83 9A 002E 132 BLSSU 30$
04 50 06 E1 0035 133 MOVL #CLIS_IVCHAR,R4
00 50 05 E5 0038 134 MOVZBL (R3)+,R0
BB AF 6E 50 3A 003C 135 BBC #6,R0,25$
14 13 0040 136 BBCC #5,R0,25$
50 6E 50 C3 0040 137 25$:
54 00038088 8F D0 0040 138 LOCC R0,(SP),CVTABL
55 50 6E 55 7A 0045 139 BEQL 30$
56 D5 0052 140 SUBL3 R0,(SP),R0
56 C8 13 004B 141 MOVL #CLIS_IVVALU,R4
50 54 7D 0058 142 EMUL R5,(SP),R0,R5
0074 8F BA 005E 143 TSTL R6
05 0062 144 BEQL 20$
0063 145 30$:
0063 146 MOVQ R4,R0
147 POPR #*M<R2,R4,R5,R6>
148 RSB
149
150 .END

```

```

: GET RADIX
: SAVE REGISTERS AND RADIX
: CLEAR RESULT
: ASSUME SUCCESS
: DECREMENT BYTE COUNT
: IF LSSU DONE
: ASSUME INVALID CHARACTER
: GET NEXT CHARACTER
: BRANCH IF 0 - 9
: CONVERT TO UPPER CASE ALPHA
: SEARCH FOR NUMERIC CHARACTER
: IF EQL INVALID CHARACTER
: CONVERT INDEX TO NUMBER
: ASSUME OVERFLOW
: COMPUTE BINARY VALUE
: TEST FOR OVERFLOW
: IF EQL CONTINUE
: GET STATUS AND RESULT
: RESTORE REGISTERS
:

```

CNVCLIATB
Symbol table

- CONVERT ASCII TO BINARY

H 5

15-SEP-1984 23:37:56
4-SEP-1984 23:15:11

VAX/VMS Macro V04-00
[CLIUTL.SRC]CNVCLIATB.MAR;1

Page 5
(1)

CLIS_IVCHAR	=	00038050		
CLIS_IVVALU	=	00038088		
CLIS-NORMAL	=	00030001		
CVTABL		00000000	R	02
LIBSCVTCLIDTB		00000010	RG	02
LIBSCVTCLIHTB		0000001A	RG	02
LIBSCVTCLIOTB		00000015	RG	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			
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE			
CNVCLIATB	00000063 (99.)	02 (2.)	NOPIC USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	13	00:00:00.09	00:00:01.13
Command processing	118	00:00:00.88	00:00:03.37
Pass 1	167	00:00:02.26	00:00:08.13
Symbol table sort	0	00:00:00.15	00:00:00.28
Pass 2	51	00:00:00.48	00:00:02.18
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	4	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	359	00:00:03.91	00:00:15.16

The working set limit was 750 pages.
 9141 bytes (18 pages) of virtual memory were used to buffer the intermediate code.
 There were 10 pages of symbol table space allocated to hold 157 non-local and 4 local symbols.
 150 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
-\$255\$DUA28:[CLIUTL.OBJ]CLIUTL.MLB;1	0
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	4

217 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:CNVCLIATB/OBJ=OBJ\$:CNVCLIATB MSRC\$:CNVCLIATB/UPDATE=(ENH\$:CNVCLIATB)+EXECMLS/LIB+LIB\$:CLIUTL/LIB

BCPRSDEF
REQ

CNVCLTATB
LIS

INFO
LIS

TYPE
REQ

CHR SUB
LIS

CNVCLINUM
LIS

SHODEVDEF
REQ

CLT MAC
MAR

CNVCLIFRM
LIS

DIGRAMS
LIS

CALC MAX
LIS

CLTUTL MAC
MAR

CUT TIME
LIS

BCMDPRS
LIS

SHOWDEF
REQ

CREATE
LIS