

VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	


```
CCCCCCCCC VV VV TTTTTTTTTT AAAAAA TTTTTTTTTT BBBB BBBB
CCCCCCCCC VV VV TTTTTTTTTT AAAAAA TTTTTTTTTT BBBB BBBB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CC          VV VV VV TT AA AA TT BB BB
CCCCCCCCC VV VV TTTTTTTTTT AAAAAA TTTTTTTTTT BBBB BBBB
CCCCCCCCC VV VV TTTTTTTTTT AAAAAA TTTTTTTTTT BBBB BBBB
BBB BBBB
```

```
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LL          IIIIIII SSSSSSSS
LLLLLLLLLLL IIIIIII SSSSSSSS
LLLLLLLLLLL IIIIIII SSSSSSSS
```


(1) 53
(1) 71

HISTORY
DECLARATIONS

; DETAILED


```
0000 1 .TITLE LIB$CVT_ATB - ASCII NUMBER TO BINARY CONVERSION
0000 2 .IDENT 'V04-000'
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 EQUATED SYMBOLS:
0000 30
0000 31 COUNT = 4 ; COUNT ARG IN ARG LIST
0000 32 STRING = 8 ; STRING ARG
0000 33 RESULT = 12 ; RESULT ADDRESS
0000 34
0000 35 ++
0000 36
0000 37 FACILITY: SYSTEM LIBRARY
0000 38
0000 39 ABSTRACT:
0000 40
0000 41 THIS ROUTINE PERFORMS ASCII INTEGER TO BINARY CONVERSION IN
0000 42 DECIMAL, OCTAL, AND HEX RADIX; RADIX DETERMINED BY ENTRY POINT.
0000 43 AN OPTIONAL LEADING SIGN IS ACCEPTED.
0000 44
0000 45 ENVIRONMENT:
0000 46
0000 47 STAR NATIVE MODE PROCESSOR, ANY ACCESS LEVEL. NO SPECIAL
0000 48 INSTRUCTIONS OR SYSTEM SERVICES ARE USED. 9 LONGWORDS OF
0000 49 STACK SPACE NEEDED.
0000 50
0000 51 --
0000 52
0000 53 .SBTTL HISTORY ; DETAILED
0000 54
0000 55 AUTHOR: ANDREW C. GOLDSTEIN 26-JAN-78 16:48
0000 56
0000 57 MODIFIED BY:
```

00000004
00000008
0000000C

LIB\$CVT_ATB
V04-000

- ASCII NUMBER TO BINARY CONVERSION K 4
HISTORY ; DETAILED

16-SEP-1984 02:17:13 VAX/VMS Macro V04-00
5-SEP-1984 04:39:03 [VMSLIB.SRC]CVTATB.MAR;1

Page 2
(1)

0000	58	:	
0000	59	:	V03-001 SBL3001 Steven B. Lionel, 28-Oct-1982
0000	60	:	Use .ENTRY instead of .WORD for entry mask.
0000	61	:	
0000	62	:	1-009 ACG0049 Andrew C. Goldstein, 20-Jun-1979 15:11
0000	63	:	Change module name to LIB\$CVT_ATB per RTL standards
0000	64	:	
0000	65	:	V0008 ACG0024 Andrew C. Goldstein, 27-Feb-1979 16:38
0000	66	:	Fix ident and PSECTs for new RTL standards
0000	67	:	
0000	68	:	**
0000	69	:	

```
0000 71 .SBTTL DECLARATIONS
0000 72 :++
0000 73 :
0000 74 : FUNCTIONAL DESCRIPTION:
0000 75 :
0000 76 : THE ROUTINE WORKS IN THE OBVIOUS MANNER OF ALL CONVERSION ROUTINES:
0000 77 : IT SCANS THE INPUT STRING AND CONVERTS EACH CHARACTER INTO ITS
0000 78 : NUMERICAL EQUIVALENT AND CHECKS IT FOR LEGALITY AGAINST THE RADIX.
0000 79 : THE NUMBER BEING ACCUMULATED IS THEN MULTIPLIED BY THE RADIX AND THE
0000 80 : NEW DIGIT IS ADDED IN. NON-RADIX CHARACTERS IN THE INPUT STRING CAUSE
0000 81 : AN ERROR RETURN. A SIGN IN OTHER THAN THE FIRST CHARACTER POSITION
0000 82 : AND OVERFLOW FROM 32 BITS (UNSIGNED) ALSO CAUSE AN ERROR RETURN.
0000 83 :
0000 84 : CALLING SEQUENCE:
0000 85 : CALL LIB$CVT_DTB (COUNT,STRING,RESULT) ; CONVERT DECIMAL TO BINARY
0000 86 : CALL LIB$CVT_OTB (COUNT,STRING,RESULT) ; CONVERT OCTAL TO BINARY
0000 87 : CALL LIB$CVT_HTB (COUNT,STRING,RESULT) ; CONVERT HEX TO BINARY
0000 88 :
0000 89 :
0000 90 : INPUT PARAMETERS:
0000 91 : COUNT: BYTE COUNT OF INPUT STRING
0000 92 : STRING: ADDRESS OF INPUT STRING
0000 93 :
0000 94 : IMPLICIT INPUTS:
0000 95 : NONE
0000 96 :
0000 97 : OUTPUT PARAMETERS:
0000 98 : RESULT: ADDRESS TO STORE LONGWORD RESULT
0000 99 :
0000 100 : IMPLICIT OUTPUTS:
0000 101 : NONE
0000 102 :
0000 103 : COMPLETION CODES:
0000 104 : R0 = 1 SUCCESSFUL CONVERSION
0000 105 : = 0 ILLEGAL CHARACTER OR ZERO BYTE COUNT
0000 106 :
0000 107 : SIDE EFFECTS:
0000 108 : NONE
0000 109 :
0000 110 : --
0000 111 :
0000 112 : .ENABL LSB
0000 113 :
00000000 114 : .PSECT _LIB$CODE,NOWRT,PIC,SHR,LONG
0000 115 :
0000 116 : BASE AND VALUE TABLES TO CONVERT DIGITS INTO THEIR NUMERICAL VALUES
0000 117 :
30 41 61 0000 118 BASETAB: .BYTE ^A'a', ^A'A', ^A'0'
39 5A 7A 0003 119 TOPTAB: .BYTE ^A'z', ^A'Z', ^A'9'
30 37 57 0006 120 VALTAB: .BYTE ^A'a'-10, ^A'A'-10, ^A'0'-00
0009 121
0009 122
0009 123 .ENTRY LIB$CVT_DTB,- ; ENTRY FOR DECIMAL CONVERT
52 0A 003C 0009 124 ^M<R2,R3,R4,R5> ; SAVE REGISTERS 2-5
OC 0C 000B 125 #10,R2 ; R2 = RADIX
000E 126 10$
0010 127
```



```
003C 0010 128 .ENTRY LIB$CVT_OTB,- ; ENTRY FOR OCTAL CONVERT
      0012 129      ^M<R2,R3,R4,R5> ; SAVE REGISTERS 2-5
52 08 D0 0012 130      MOVL #8,R2 ; R2 = RADIX
    05 11 0015 131      BRB 10$
      0017 132
003C 0017 133 .ENTRY LIB$CVT_HTB,- ; ENTRY FOR HEX CONVERT
      0019 134      ^M<R2,R3,R4,R5> ; SAVE REGISTERS 2-5
52 10 D0 0019 135      MOVL #16,R2 ; R2 = RADIX
      001C 136
      50 7C 001C 137 10$: 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$: AOBLS #3,R1,40$ ; 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$: AOBLS COUNT(AP),R3,20$ ; 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
OC 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
```


LIB\$CVT_ATB
Symbol Table- ASCII NUMBER TO BINARY CONVERSION^{N 4}16-SEP-1984 02:17:13
5-SEP-1984 04:39:03VAX/VMS Macro V04-00
[VMSLIB.SRC]CVTATB.MAR;1Page 5
(1)

BASETAB	= 00000000	R	02	OP\$-CVTLD	= 0000006E	OP\$-SCANC	= 0000002A
COUNT	= 00000004			OP\$-CVTLF	= 0000004E	OP\$-SKPC	= 0000003B
LIB\$CVT_DTB	= 00000009	RG	02	OP\$-CVTLG	= 00004EFD	OP\$-SPANC	= 0000002B
LIB\$CVT_HTB	= 00000017	RG	02	OP\$-CVTLH	= 00006EFD	OP\$-SUBD2	= 00000062
LIB\$CVT_OTB	= 00000010	RG	02	OP\$-CVTLP	= 000000F9	OP\$-SUBD3	= 00000063
OP\$-ACBD	= 0000006F			OP\$-CVTPL	= 00000036	OP\$-SUBF2	= 00000042
OP\$-ACBF	= 0000004F			OP\$-CVTPS	= 00000008	OP\$-SUBF3	= 00000043
OP\$-ACBG	= 00004FFD			OP\$-CVTPT	= 00000024	OP\$-SUBG2	= 000042FD
OP\$-ACBH	= 00006FFD			OP\$-CVTRDL	= 0000006B	OP\$-SUBG3	= 000043FD
OP\$-ADDD2	= 00000060			OP\$-CVTRFL	= 0000004B	OP\$-SUBH2	= 000062FD
OP\$-ADDD3	= 00000061			OP\$-CVTRGL	= 00004BFD	OP\$-SUBH3	= 000063FD
OP\$-ADDF2	= 00000040			OP\$-CVTRHL	= 00006BFD	OP\$-SUBP4	= 00000022
OP\$-ADDF3	= 00000041			OP\$-CVTSP	= 00000009	OP\$-SUBP6	= 00000023
OP\$-ADDG2	= 000040FD			OP\$-CVTTP	= 00000026	OP\$-TSTD	= 00000073
OP\$-ADDG3	= 000041FD			OP\$-CVTWD	= 0000006D	OP\$-TSTF	= 00000053
OP\$-ADDH2	= 000060FD			OP\$-CVTWF	= 0000004D	OP\$-TSTG	= 000053FD
OP\$-ADDH3	= 000061FD			OP\$-CVTWG	= 00004DFD	OP\$-TSTH	= 000073FD
OP\$-ADDP4	= 00000020			OP\$-CVTWH	= 00006DFD	RESULT	= 0000000C
OP\$-ADDP6	= 00000021			OP\$-DIVD2	= 00000066	STRING	= 00000008
OP\$-ASHP	= 000000F8			OP\$-DIVD3	= 00000067	TOPTAB	00000003 R 02
OP\$-CLRD	= 0000007C			OP\$-DIVF2	= 00000046	VALTAB	00000006 R 02
OP\$-CLRF	= 000000D4			OP\$-DIVF3	= 00000047		
OP\$-CLRG	= 0000007C			OP\$-DIVG2	= 000046FD		
OP\$-CLRH	= 00007CFD			OP\$-DIVG3	= 000047FD		
OP\$-CMPD	= 00000071			OP\$-DIVH2	= 000066FD		
OP\$-CMPF	= 00000051			OP\$-DIVH3	= 000067FD		
OP\$-CMPG	= 000051FD			OP\$-DIVP	= 00000027		
OP\$-CMPH	= 000071FD			OP\$-EDITPC	= 00000038		
OP\$-CMPP3	= 00000035			OP\$-EMODD	= 00000074		
OP\$-CMPP4	= 00000037			OP\$-EMODF	= 00000054		
OP\$-CRC	= 0000000B			OP\$-EMODG	= 000054FD		
OP\$-CVTBD	= 0000006C			OP\$-EMODH	= 000074FD		
OP\$-CVTBF	= 0000004C			OP\$-MATCHC	= 00000039		
OP\$-CVTBG	= 00004CFD			OP\$-MNEGD	= 00000072		
OP\$-CVTBH	= 00006CFD			OP\$-MNEGF	= 00000052		
OP\$-CVTDB	= 00000068			OP\$-MNEGG	= 000052FD		
OP\$-CVTDF	= 00000076			OP\$-MNEGH	= 000072FD		
OP\$-CVTDH	= 000032FD			OP\$-MOVD	= 00000070		
OP\$-CVTDL	= 0000006A			OP\$-MOVF	= 00000050		
OP\$-CVTDW	= 00000069			OP\$-MOVG	= 000050FD		
OP\$-CVTFB	= 00000048			OP\$-MOVH	= 000070FD		
OP\$-CVTFD	= 00000056			OP\$-MOVV	= 00000034		
OP\$-CVTFG	= 000099FD			OP\$-MOVTC	= 0000002E		
OP\$-CVTFH	= 000098FD			OP\$-MOVTUC	= 0000002F		
OP\$-CVTFL	= 0000004A			OP\$-MULD2	= 00000064		
OP\$-CVTFW	= 00000049			OP\$-MULD3	= 00000065		
OP\$-CVTGB	= 000048FD			OP\$-MULF2	= 00000044		
OP\$-CVTGF	= 000033FD			OP\$-MULF3	= 00000045		
OP\$-CVTGH	= 000056FD			OP\$-MULG2	= 000044FD		
OP\$-CVTGL	= 00004AFD			OP\$-MULG3	= 000045FD		
OP\$-CVTGW	= 000049FD			OP\$-MULH2	= 000064FD		
OP\$-CVTHB	= 000068FD			OP\$-MULH3	= 000065FD		
OP\$-CVTHD	= 0000F7FD			OP\$-MULP	= 00000025		
OP\$-CVTHF	= 0000F6FD			OP\$-POLYD	= 00000075		
OP\$-CVTHG	= 000076FD			OP\$-POLYF	= 00000055		
OP\$-CVTHL	= 00006AFD			OP\$-POLYG	= 000055FD		
OP\$-CVTHW	= 000069FD			OP\$-POLYH	= 000075FD		

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR
_LIB\$CODE	0000007A (122.)	02 (2.)	PIC USR

CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.09	00:00:00.52
Command processing	124	00:00:00.72	00:00:04.92
Pass 1	344	00:00:08.49	00:00:16.90
Symbol table sort	0	00:00:00.52	00:00:01.12
Pass 2	48	00:00:03.09	00:00:06.38
Symbol table output	11	00:00:00.10	00:00:00.30
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	564	00:00:13.03	00:00:30.16

The working set limit was 1200 pages.
36819 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 18 object records in Pass 2.
134 pages of virtual memory were used to define 133 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4

420 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/DISA=TRACE/LIS=LISS:CVTATB/OBJ=OBJ\$:CVTATB MASD\$:[EMULAT.SRC]MISSING/UPDATE=(MASD\$:[EMULAT.ENH]MISSING)+MASD\$:[VMSLIB.SRC]CVTA

0435 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

