

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	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL

```

CCCCCCCC NN      NN  VV      VV  CCCCCCCC LL      IIIIII  FFFFFFFF  RRRRRRRR  MM      MM
CCCCCCCC NN      NN  VV      VV  CCCCCCCC LL      IIIIII  FFFFFFFF  RRRRRRRR  MM      MM
CC        NN      NN  VV      VV  CC        LL      II      FF        RR      RR  MMMM  MMMM
CC        NN      NN  VV      VV  CC        LL      II      FF        RR      RR  MMMM  MMMM
CC        NNNN    NN  VV      VV  CC        LL      II      FF        RR      RR  MM  MM  MM
CC        NNNN    NN  VV      VV  CC        LL      II      FF        RR      RR  MM  MM  MM
CC        NN  NN  NN  VV      VV  CC        LL      II      FFFFFFFF  RRRRRRRR  MM      MM
CC        NN  NN  NN  VV      VV  CC        LL      II      FFFFFFFF  RRRRRRRR  MM      MM
CC        NN      NNNN  VV      VV  CC        LL      II      FF        RR  RR  MM      MM
CC        NN      NNNN  VV      VV  CC        LL      II      FF        RR  RR  MM      MM
CC        NN      NN  VV      VV  CC        LL      II      FF        RR      RR  MM      MM
CC        NN      NN  VV      VV  CC        LL      II      FF        RR      RR  MM      MM
CC        NN      NN  VV      VV  CC        LL      II      FF        RR      RR  MM      MM
CC        NN      NN  VV      VV  CC        LL      II      FF        RR      RR  MM      MM
CC        NN      NN  VV      VV  CC        LL      II      FF        RR      RR  MM      MM
CC        NN      NN  VV      VV  CC        LL      II      FF        RR      RR  MM      MM
CCCCCCCC NN      NN  VV      VV  CCCCCCCC LLLLLLLLLL IIIIII  FF        RR      RR  MM      MM
CCCCCCCC NN      NN  VV      VV  CCCCCCCC LLLLLLLLLL IIIIII  FF        RR      RR  MM      MM

```

```

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

```

(2) 49  
(3) 78

DECLARATIONS  
CONVERT CLI FORMS AND CHARACTERISTICS

```
0000 1 .TITLE CNVCLIFRM - CONVERT COMMAND INPUT FORMS TYPE
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 :++
0000 30 : FACILITY: CLI UTILITIES
0000 31
0000 32 : ABSTRACT: THIS ROUTINE CONVERTS A FORMS/CHARACTERISTIC TYPE
0000 33 : INTO A BINARY VALUE.
0000 34
0000 35
0000 36 : ENVIRONMENT: NATIVE MODE, NON-PRIVILEGED
0000 37
0000 38 : AUTHOR: STEVE BECKHARDT, CREATION DATE: 13-FEB-78
0000 39
0000 40 : M ED BY:
0000 41
0000 42 : V03 GWF0107 GARY FOWLER 21-SEP-1981
0000 43 : Change logical name used in opening files to SYSSMANAGER
0000 44
0000 45 : V02 LMK0001 LEN KAWELL 15-FEB-1980
0000 46 : ADD DE-QUEUE CHARACTERISTICS CONVERSION.
0000 47 :--
```

```
0000 49 .SBTTL DECLARATIONS
0000 50 :
0000 51 : INCLUDE FILES:
0000 52 :
0000 53 :
0000 54 : MACROS:
0000 55 :
0000 56 :
0000 57 :
0000 58 : EQUATED SYMBOLS:
0000 59 :
0000 60 $CLIMSGDEF ; DEFINE CLI ERROR CODES
0000 61 $FABDEF ; FAB DEFINITIONS
0000 62 $RABDEF ; RAB DEFINITIONS
0000 63
0000087 0000 64 RCBFSZ=135 ; RECORD BUFFER SIZE FOR FORMS
0000 65 ; DEFINITION FILE
0000 66
0000 67 :
0000 68 : OWN STORAGE:
0000 69 :
0000 70 PURE_SECTION PURE_CLIUTL
0000 71
0000 72
52 45 47 41 4E 41 4D 24 53 59 53 00' 0000 73 FFILNAM: .ASCIC /SYSS$MANAGER:FORMSTYPE.DAT/ ; FILE NAME OF FORMS DEFINITION FILE
44 2E 45 50 59 54 53 4D 52 4F 46 3A 000C
54 41 0018
19 0000
52 45 47 41 4E 41 4D 24 53 59 53 00' 001A 74 CFILNAM: .ASCIC /SYSS$MANAGER:CHARTYPE.DAT/ ; FILE NAME OF CHARACTERISTICS DEFINITION FILE
41 44 2E 45 50 59 54 52 41 48 43 3A 0026
54 0032
18 001A
0033 75
0033 76
```

```

0033 78 .SBTTL CONVERT CLI FORMS AND CHARACTERISTICS
0033 79 :++
0033 80 : FUNCTIONAL DESCRIPTION:
0033 81 :
0033 82 : THESE ROUTINES ARE CALLED TO CONVERT THE FORMS/CHARACTERISTICS
0033 83 : ENTERED ON A COMMAND INTO A BINARY BYTE VALUE.
0033 84 : THIS WILL CONVERT EITHER A NUMBER DIRECTLY, OR
0033 85 : A ALPHANUMERIC NAME. ALPHANUMERIC FORMS NAMES ARE
0033 86 : STORED IN FILE "SYSSMANAGER:FORMSTYPE.DAT" AND CHARACTERISTICS
0033 87 : NAMES ARE STORED IN "SYSSMANAGER:CHARTYPE.DAT.
0033 88 : FORMS/CHARACTERISTICS DEFINITIONS IN THE FILE MUST BE IN
0033 89 : THE FOLLOWING FORMAT:
0033 90 : % TYPE NUMBER
0033 91 : THE PERCENT SIGN MUST BE THE FIRST CHARACTER IN THE RECORD.
0033 92 :
0033 93 : CALLING SEQUENCE:
0033 94 :
0033 95 : VIA "CALL" INSTRUCTION
0033 96 :
0033 97 : INPUT PARAMETERS:
0033 98 :
0033 99 : 4(AP) IS THE ADDRESS OF A QUAD WORD DESCRIPTOR FOR THE TYPE CODE
0033 100 : 8(AP) IS THE ADDRESS OF BYTE TO STORE RESULT(OPTION)
0033 101 :
0033 102 : IMPLICIT INPUTS:
0033 103 :
0033 104 : NONE
0033 105 :
0033 106 : OUTPUT PARAMETERS:
0033 107 :
0033 108 : THE CONVERTED FORMS/CHAR TYPE IS RETURNED IN R1 AND AT THE LOCATION
0033 109 : SPECIFIED IF THE CALL ARGUMENT LIST HAD MORE THAN ONE ARGUMENT.
0033 110 :
0033 111 : IMPLICIT OUTPUTS:
0033 112 :
0033 113 : CONVERTED VALUE IN R1
0033 114 :
0033 115 : COMPLETION CODES:
0033 116 :
0033 117 : R0 IS SET TRUE OR FALSE DEPENDING UPON SUCCESS OR FAILURE OF CONVERSION
0033 118 :
0033 119 : SIDE EFFECTS:
0033 120 :
0033 121 : NONE
0033 122 :
0033 123 :--
0033 124 : PURE_SECTION PURE_CLIUTL
0033 125 :
0033 126 LIBSCNVCLIFORMS:: : CONVERT FORMS TYPE
0033 127 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
56 C8 AF OFFC 0033 127
06 9E 0035 128 MOVAB FFILNAM,R6 : SET ADDRESS OF FORMS FILE
0039 129 BRB CONVERT :
003B 130
003B 131 LIBSCNVCLIQCIAR:: : CONVERT CHARACTERISTICS TYPE
56 DA AF OFFC 003B 132 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
003D 133 MOVAB CFILNAM,R6 : SET ADDRESS OF CHAR FILENAME
0041 134

```

				0041	135	CONVERT:			
				0041	136				; CONVERT TYPE
				0041	137				; FIRST SEE IF WE CAN CONVERT A NUMBER
				0044	138	PUSHAQ	@4(AP)		; POINTER TO TOKEN DESCRIPTOR
				004B	139	CALLS	#1,LIB\$CVT_DECBIN		; TRY TO CONVERT IT
				004E	140	BLBC	RO,5\$		; CONVERSION FAILED
				0051	141	BRW	60\$		; SUCCESS! VALUE IN R1
				0051	142				
				0051	143	5\$:			; NUMBER CONVERSION FAILED. TRY LOOKING IT UP IN FILE
				0055	144	MOVQ	@4(AP),R10		; PUT TOKEN DESCRIPTOR IN R10,R11
				0055	145				
				0055	146				; ALLOCATE SPACE FOR FAB, RAB, AND BUFFERS ON STACK
				0059	147	MOVAB	-FAB\$C_BLN(SP),R7		; R7 POINTS TO FAB
				005D	148	MOVAB	-RAB\$C_BLN(R7),R8		; R8 POINTS TO RAB
				0062	149	MOVAB	-<RCBFSZ+1>(R8),R9		; R9 POINTS TO RECORD BUFFER
				0065	150	MOVL	R9,SP		; SET STACK POINTER
				0065	151				
				0065	152				; CLEAR FAB AND RAB
				006D	153	MOVCS	#0,(R8),#0,#RAB\$C_BLN+FAB\$C_BLN,(R8)		
				006D	154				
				006D	155				; SET VARIOUS FAB FIELDS
				006D	156	ASSUME	FAB\$B_BLN EQ FAB\$B_PID+1		
				0072	157	MOVW	#<FAB\$C_BLN@8>+FAB\$C_BID,FAB\$B_BID(R7)		; BLOCK ID AND LENGTH
				0076	158	MOVW	#FAB\$M_GET,FAB\$B_FACT(R7)		; FILE ACCESS
				007A	159	MOVW	(R6)+,FAB\$B_FNS(R7)		; FILE NAME SIZE
				007E	160	MOVAB	(R6),FAB\$B_FNA(R7)		; FILE NAME ADDRESS
				007E	161				
				007E	162				; OPEN FILE
				0087	163	\$OPEN	FAB=(R7)		
				008A	164	BLBS	RO,15\$		
				008D	165	BRW	80\$		; SUCCESSFUL OPEN
				008D	166	10\$:			; ERROR (ASSUME FILE NOT FOUND)
				008D	167				
				008D	168	15\$:			; SET VARIOUS RAB FIELDS
				0092	169	ASSUME	RAB\$B_BLN EQ RAB\$B_BID+1		
				0096	170	MOVW	#<RAB\$C_BLN@8>+RAB\$C_BID,RAB\$B_BID(R8)		; BLOCK ID AND LENGTH
				009A	171	MOVAB	(R7),RAB\$B_FAB(R8)		; FAB ADDRESS
				009E	172	MOVW	#RAB\$C_SEQ,RAB\$B_RAC(R8)		; RECORD ACCESS
				00A4	173	MOVAB	(R9),RAB\$B_UBF(R8)		; USER BUFFER ADDRESS (ON STACK)
				00A4	174	MOVW	#RCBFSZ,RAB\$B_USZ(R8)		; USER BUFFER SIZE
				00AD	175				
				00B0	176	\$CONNECT	RAB=(R8)		
				00B0	177	BLBC	RO,70\$		; CONNECT ERROR
				00B0	178				
				00B9	179	20\$:			; READ A LINE FROM FILE
				00BC	180	\$GET	RAB=(R8)		
				00BC	181	BLBC	RO,70\$		; ERROR (ASSUME END OF FILE)
				00BC	182				
				00C0	183				; GET LENGTH OF RECORD AND STORE A 0 AT THE END (FOR PARSING ROUTINES)
				00C3	184	MOVZWL	RAB\$W_RSZ(R8),RO		; RECORD SIZE
				00C3	185	CLRB	(R9)[RO]		; WE NOW HAVE AN ASCIZ STRING
				00C3	186				
				00C6	187				; IS THIS LINE INTERESTING? (DOES IT BEGIN WITH %)
				00C9	188	MOVL	R9,R6		; POINTER TO BUFFER
				00CB	189	CMPB	(R6)+,#^A/%		
				00CB	190	BNEQ	20\$		; NOT OF INTEREST. GET NEXT LINE
				00CB	191				
				00CB	191				; NOW GET FIRST TOKEN ON LINE AND SEE IF IT MATCHES INPUT SYMBOL
				00CB	191	BSBW	CHR\$GETOKEN		; RETURNS TOKEN DESCRIPTOR IN R3,R4

```

64 53 00 6B E0 13 00CE 192 BEQL 20$ ; NULL TOKEN; GET NEXT LINE
5A 2D 00D0 193 CMPCS R10,(R11),#0,R3,(R4) ; IT IS CONSIDERED A MATCH IF THE INPUT
50 B5 00D6 194 TSTW R0 ; SYMBOL IS AN INITIAL SUBSTRING OF THE
; TOKEN (R0=0)
D6 12 00D8 195 BNEQ 20$ ; NO MATCH; GET NEXT LINE
; OODA 196
; OODA 197
; OODA 198 ; HAVE A MATCH. R6 POINTS TO NEXT TOKEN (HOPEFULLY VALUE)
FF23' 30 00DA 199 BSBW CHR$GETOKEN
30 13 00DD 200 BEQL 70$ ; NULL TOKEN --> ERROR
; OODF 201
; OODF 202 ; HAVE TOKEN. SET UP TO CALL ROUTINE TO CONVERT TO BINARY
7E 53 7D 00DF 203 MOVQ R3,-(SP) ; PUT TOKEN DESCRIPTOR ON STACK
; OOE2 204
; OOE2 205 ; BUILD ARGUMENT LIST ON STACK
00000000'EF 6E 7F 00E2 206 PUSHAQ (SP) ; POINTER TO TOKEN DESCRIPTOR
01 FB 00E4 207 CALLS #1,LIB$CVT_DECBIN ; CONVERT TO BINARY
8E 7C 00EB 208 CLRQ (SP)+ ; REMOVE DESCRIPTOR
1F 50 E9 00ED 209 BLBC R0,70$ ; CONVERSION ERROR
; OOF0 210
; OOF0 211 50$: ; SUCCESS RETURN (WITH CLOSE). VALUE IS IN R1
51 DD 00F0 212 PUSHL R1 ; SAVE R1 ACROSS CLOSE
00F2 213 $CLOSE FAB=(R7) ; CLOSE FILE
02 BA 00FB 214 POPR #^M<R1> ; RESTORE R1
; OOFD 215
; OOFD 216 60$: ; SUCCESS RETURN (WITHOUT CLOSE). VALUE IS IN R1.
50 00030001 8F DO 00FD 217 MOVL #CLIS_NORMAL,R0 ; INDICATE SUCCESS
01 6C D1 0104 218 CMPL (AP),#1 ; STORE IN MEMORY IF MORE THAN ONE ARG
08 BC 16 1B 0107 219 BLEQU 90$ ; DON'T STORE
51 90 0109 220 MOVB R1,@8(AP) ; STORE
10 11 010D 221 BRB 90$
; O10F 222
; O10F 223 70$: ; FAILURE RETURN (WITH CLOSE)
010F 224 $CLOSE FAB=(R7) ; CLOSE FILE
; O118 225
; O118 226 80$: ; FAILURE RETURN (WITHOUT CLOSE)
50 00038832 8F DO 0118 227 MOVL #CLIS_VALCNVERR,R0 ; INDICATE CONVERSION ERROR
; O11F 228
; O11F 229 90$: RET
0120 230
0120 231
0120 232
0120 233 .END

```

CNVCLIFRM  
Symbol table

- CONVERT COMMAND INPUT FORMS TYPE C 6

```

SS.TMP1      = 00000001
SS.TMP2      = 00000067
CFILNAM      = 0000001A R    02
CHR$GETOKEN  ***** X    02
CLIS_NORMAL  = 00030001
CLIS_VALCNVERR = 00038832
CONVERT      = 00000041 R    02
FABS$B_BID   = 00000000
FABS$B_BLN   = 00000001
FABS$B_FAC   = 00000016
FABS$B_FNS   = 00000034
FABS$C_BID   = 00000003
FABS$C_BLN   = 00000050
FABS$L_FNA   = 0000002C
FABS$M_GET   = 00000002
FFILNAM      = 00000000 R    02
LIB$CNVCLIFORMS = 00000033 RG   02
LIB$CNVCLIQCHAR = 0000003B RG   02
LIB$CVT_DECBIN ***** X    02
RABS$B_BID   = 00000000
RABS$B_BLN   = 00000001
RABS$B_RAC   = 0000001E
RABS$C_BID   = 00000001
RABS$C_BLN   = 00000044
RABS$C_SEQ   = 00000000
RABS$L_FAB   = 0000003C
RABS$L_UBF   = 00000024
RABS$W_RSZ   = 00000022
RABS$W_USZ   = 00000020
RCBFSZ      = 00000087
SYSS$CLOSE   ***** GX   02
SYSS$CONNECT ***** GX   02
SYSS$GET     ***** GX   02
SYSS$OPEN    ***** GX   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			
PURE_CLIUTL	00000120 ( 288.)	02 ( 2.)	NOPIC USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	16	00:00:00.07	00:00:00.64
Command processing	109	00:00:00.81	00:00:06.07
Pass 1	200	00:00:05.06	00:00:19.86
Symbol table sort	0	00:00:00.50	00:00:02.49
Pass 2	62	00:00:01.04	00:00:03.70
Symbol table output	4	00:00:00.05	00:00:00.05

Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	395	00:00:07.57	00:00:32.85

The working set limit was 1200 pages.  
26652 bytes (53 pages) of virtual memory were used to buffer the intermediate code.  
There were 30 pages of symbol table space allocated to hold 465 non-local and 9 local symbols.  
233 source lines were read in Pass 1, producing 13 object records in Pass 2.  
18 pages of virtual memory were used to define 16 macros.

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

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[CLIUTL.OBJ]CLIUTL.MLB;1	1
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	12
TOTALS (all libraries)	13

590 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

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

