

DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

```
DDDDDDDD  EEEEEEEEE  SSSSSSSS  CCCCCCCC  RRRRRRRR  VV      VV  AAAAAA  LL
DDDDDDDD  EEEEEEEEE  SSSSSSSS  CCCCCCCC  RRRRRRRR  VV      VV  AAAAAA  LL
DD      DD  EE      SS      CC      RR      RR  VV      VV  AA      AA  LL
DD      DD  EE      SS      CC      RR      RR  VV      VV  AA      AA  LL
DD      DD  EE      SS      CC      RR      RR  VV      VV  AA      AA  LL
DD      DD  EE      SS      CC      RR      RR  VV      VV  AA      AA  LL
DD      DD  EEEEEEEE  SSSSSS  CC      RR      RR  VV      VV  AA      AA  LL
DD      DD  EEEEEEEE  SSSSSS  CC      RR      RR  VV      VV  AA      AA  LL
DD      DD  EE      SS      CC      RR      RR  VV      VV  AAAAAAAAAA  LL
DD      DD  EE      SS      CC      RR      RR  VV      VV  AAAAAAAAAA  LL
DD      DD  EE      SS      CC      RR      RR  VV      VV  AA      AA  LL
DD      DD  EE      SS      CC      RR      RR  VV      VV  AA      AA  LL
DDDDDDDD  EEEEEEEEE  SSSSSSSS  CCCCCCCC  RR      RR  VV      VV  AA      AA  LLLLLLLLLL
DDDDDDDD  EEEEEEEEE  SSSSSSSS  CCCCCCCC  RR      RR  VV      VV  AA      AA  LLLLLLLLLL
.....
.....
.....
.....
```

```
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)	56	GET RESULT PARSE DESCRIPTOR VALUES
(3)	102	GET KEYWORD NUMBER
(4)	124	SORT THE TOKEN DESCRIPTOR TABLE

```
0000 1 .TITLE DESCRVAL - TOKEN TABLE MANIPULATION ROUTINES
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 D. N. CUTLER 24-APR-77
0000 29
0000 30 PARSE TABLE DESCRIPTOR VALUE MANIPULATION ROUTINES
0000 31
0000 32 MODIFIED BY:
0000 33
0000 34 V03-004 PCG0004 Peter George 27-May-1983
0000 35 Read DCL$SORT_TOKENS.
0000 36
0000 37 V03-003 PCG0003 Peter George 06-May-1983
0000 38 Move DCL$SORT_TOKENS to PARSENT.
0000 39
0000 40 V03-002 PCG0002 Peter George 30-Apr-1983
0000 41 Add DCL$SORT_TOKENS.
0000 42
0000 43 V03-001 PCG0001 Peter George 01-Mar-1983
0000 44 Add DCL$GETNVAL.
0000 45 ---
0000 46
0000 47
0000 48 MACRO LIBRARY CALLS
0000 49
0000 50
0000 51 WRKDEF ;DEFINE COMMAND WORK AREA
0000 52 PTRDEF ;DEFINE TOKEN DESCRIPTOR
0000 53
00000000 54 .PSECT DCL$ZCODE,BYTE,RD,NOWRT
```



```
0000 56 .SBTTL GET RESULT PARSE DESCRIPTOR VALUES
0000 57 :+
0000 58 : DCL$GETDVAL - GET RESULT PARSE DESCRIPTOR VALUES
0000 59 :
0000 60 : THIS ROUTINE IS CALLED TO OBTAIN THE VALUES FROM THE NEXT DESCRIPTOR IN
0000 61 : THE RESULT PARSE TABLE.
0000 62 :
0000 63 : INPUTS:
0000 64 :
0000 65 :     WRK_L_RSLNXT = NEXT TOKEN DESCRIPTOR TO PROCESS
0000 66 :
0000 67 : OUTPUTS:
0000 68 :
0000 69 :     R0 LOW BIT CLEAR INDICATES PARSE DESCRIPTOR TABLE IS EMPTY.
0000 70 :
0000 71 :     R0 LOW BIT SET INDICATES NEXT PARSE DESCRIPTOR HAS BEEN OBTAINED
0000 72 :     WITH:
0000 73 :
0000 74 :     R1 = VALUE FIELD OF DESCRIPTOR.
0000 75 :     R2 = ADDRESS OF ITEM IN COMMAND BUFFER.
0000 76 :     R3 = FLAGS FIELD OF DESCRIPTOR.
0000 77 :     R4 = FIELD TERMINATOR CLASS NUMBER.
0000 78 :     R5 = TYPE OF ENTRY.
0000 79 : -
0000 80 :
0000 81 DCL$GETDVAL::
0000 82     CLRL R0 ;GET RESULT PARSE DESCRIPTOR VALUES
0000 83     MOVL WRK_L_RSLNXT(R10),R1 ;ASSUME TABLE IS EMPTY
0000 84     CMPL R1,WRK_L_RSLEND(R10) ;GET ADDRESS OF NEXT TOKEN DESCRIPTOR
0000 85     BGEQU 10$ ;ANY MORE ENTRIES IN TABLE?
0000 86     ADDL #PTR_C_LENGTH,WRK_L_RSLNXT(R10) ;IF GEQU NO
0000 87     EXTZV #PTR_V_OFFSET,#PTR_S_OFFSET,- ;POINT TO NEXT DESCRIPTOR IN TABLE
0000 88     PTR_C_DESCR(R1),R2 ;EXTRACT OFFSET TO ITEM
0000 89     MOVAB WRK_G_BUFFER(R10)[R2],R2 ;CALCULATE ACTUAL ITEM ADDRESS
0000 90     EXTZV #PTR_V_FLAGS,#PTR_S_FLAGS,- ;EXTRACT FLAGS
0000 91     PTR_C_DESCR(R1),R3
0000 92     EXTZV #PTR_V_TERM,#PTR_S_TERM,- ;EXTRACT TERMINATOR CLASS
0000 93     PTR_C_DESCR(R1),R4
0000 94     EXTZV #PTR_V_TYPE,#PTR_S_TYPE,- ;EXTRACT ITEM TYPE
0000 95     PTR_C_DESCR(R1),R5
0000 96     EXTZV #PTR_V_VALUE,#PTR_S_VALUE,- ;EXTRACT VALUE
0000 97     PTR_C_DESCR(R1),R0
0000 98     INCL R0 ;INDICATE NEXT DESCRIPTOR OBTAINED
0000 99     RSB
0000 100 10$:
0000 101
```

51 BA AA 50 D4 0000 82
B6 AA 51 D1 0002 83
25 1E 0006 84
BA AA 0C C0 000A 85
OC 08 EF 000C 86
52 52 61 0010 87
F492 CA42 9E 0013 88
04 14 EF 0015 89
53 61 EF 001B 90
04 18 EF 001E 91
54 61 EF 0020 92
04 1C EF 0023 93
55 61 EF 0025 94
08 00 EF 0028 95
51 61 EF 002A 96
50 D6 002D 97
05 05 002F 98
0031 99
0032 100

```
0032 102 .SBTTL GET KEYWORD NUMBER
0032 103 :+
0032 104 : DCL$GETNVAL - GET KEYWORD NUMBER
0032 105 :
0032 106 : THIS ROUTINE IS CALLED TO OBTAIN THE KEYWORD NUMBER FROM THE CURRENT
0032 107 : DESCRIPTOR IN THE RESULT PARSE TABLE.
0032 108 :
0032 109 : INPUTS:
0032 110 :
0032 111 : WRK_L_RSLNXT = NEXT TOKEN DESCRIPTOR TO PROCESS
0032 112 :
0032 113 : OUTPUTS:
0032 114 :
0032 115 : R1 = NUMBER FIELD OF DESCRIPTOR.
0032 116 :-
0032 117 :
0032 118 DCL$GETNVAL::
0032 119 SUBL3 #PTR_K LENGTH,-
51 BA AA C3 0034 120 WRK_C_RSLNXT(R10),R1
51 05 A1 9A 0037 121 MOVZBL PTR_B_NUMBER(R1),R1
05 003B 122 RSB
; GET KEYWORD NUMBER
; GET ADDRESS OF TOKEN DESCRIPTOR
; GET KEYWORD NUMBER
;
```



```
003C 124 .SBTTL SORT THE TOKEN DESCRIPTOR TABLE
003C 125
003C 126 :+ DCL$SORT_TOKENS - SORT THE TOKEN DESCRIPTOR TABLE.
003C 127
003C 128 : THIS ROUTINE IS CALLED TO SORT THE TOKEN DESCRIPTOR TABLE INTO
003C 129 : CMDQUAL, PARM, PARMQUAL ORDER TO MAKE GETDVAL PROCESSING IN INTERNAL
003C 130 : CLI ROUTINES EASIER AND TO MAKE DISALLOW PROCESSING EASIER.
003C 131
003C 132 : INPUTS:
003C 133
003C 134 : R10 = ADDRESS OF WRK DATA STRUCUTURE
003C 135
003C 136 : OUTPUTS:
003C 137
003C 138 : R0-R9 ARE DESTROYED.
003C 139
003C 140 : ---
003C 141 DCL$SORT_TOKENS::
003C 142
003C 143 : PUSH R10
003C 144 : MOVAB WRK_G_RESULT(R10), R7
003C 145 : MOVL R7, R8
003C 146 : MOVAB -WRK_C_RSLBUFSIZ(SP), SP
003C 147 : MOVL SP, R9
003C 148 10$: CMPL R7, WRK_L_RSLNXT(R10)
003C 149 : BEQL 30$
003C 150 : MOVC3 #PTR_C_LENGTH, (R7), (R9)
003C 151 : ADDL #PTR_C_LENGTH, R7
003C 152 : CMPZV #PTR_V_TYPE, #PTR_S_TYPE, -
003C 153 : (R9), #PTR_K_CMDQUAL
003C 154 : BNEQ 10$
003C 155 : MOVC5 #0, (SP), #0, #PTR_C_LENGTH, -
003C 156 : -PTR_C_LENGTH(R7)
003C 157 : ADDL #PTR_C_LENGTH, R9
003C 158 20$: CMPZV #PTR_V_TYPE, #PTR_S_TYPE, -
003C 159 : (R7), #PTR_K_QUALVALU
003C 160 : BNEQ 10$
003C 161 : MOVC3 #PTR_C_LENGTH, (R7), (R9)
003C 162 : ADDL #PTR_C_LENGTH, R9
003C 163 : MOVC5 #0, (SP), #0, #PTR_C_LENGTH, (R7)
003C 164 : ADDL #PTR_C_LENGTH, R7
003C 165 : BRB 20$
003C 166
003C 167 30$: MOVL R7, R10
003C 168 40$: CMPL R7, R8
003C 169 : BEQL 50$
003C 170 : SUBL #PTR_C_LENGTH, R7
003C 171 : MOVL R7, R8
003C 172 : TSTL (R6)
003C 173 : BEQL 40$
003C 174 : SUBL #PTR_C_LENGTH, R10
003C 175 : MOVC3 #PTR_C_LENGTH, (R6), (R10)
003C 176 : BRB 40$
003C 177 50$: CMPL R9, SP
003C 178 : BEQL 60$
003C 179 : SUBL #PTR_C_LENGTH, R10
003C 180 : SUBL #PTR_C_LENGTH, R9

57 F9B6 CA DD 003C 143 : PUSH R10
58 57 DO 003E 144 : MOVAB WRK_G_RESULT(R10), R7
5E FA00 CE 9E 0043 145 : MOVL R7, R8
59 5E DO 0046 146 : MOVAB -WRK_C_RSLBUFSIZ(SP), SP
BA AA 57 D1 004B 147 : MOVL SP, R9
31 13 0052 148 10$: CMPL R7, WRK_L_RSLNXT(R10)
69 67 OC 28 0054 149 : BEQL 30$
57 OC CO 0058 150 : MOVC3 #PTR_C_LENGTH, (R7), (R9)
04 1C ED 005B 151 : ADDL #PTR_C_LENGTH, R7
00 69 ED 005E 152 : CMPZV #PTR_V_TYPE, #PTR_S_TYPE, -
EC 12 0060 153 : (R9), #PTR_K_CMDQUAL
00 6E 00 2C 0062 154 : BNEQ 10$
F4 A7 0067 155 : MOVC5 #0, (SP), #0, #PTR_C_LENGTH, -
59 OC CO 0069 156 : -PTR_C_LENGTH(R7)
04 1C ED 006C 157 : ADDL #PTR_C_LENGTH, R9
02 67 ED 006F 158 20$: CMPZV #PTR_V_TYPE, #PTR_S_TYPE, -
DB 12 0071 159 : (R7), #PTR_K_QUALVALU
69 67 OC 28 0073 160 : BNEQ 10$
59 OC CO 0077 161 : MOVC3 #PTR_C_LENGTH, (R7), (R9)
67 OC 00 2C 007A 162 : ADDL #PTR_C_LENGTH, R9
57 OC CO 0080 163 : MOVC5 #0, (SP), #0, #PTR_C_LENGTH, (R7)
E7 11 0083 164 : ADDL #PTR_C_LENGTH, R7
5A 57 DO 0085 165 : BRB 20$
58 57 D1 0088 166 :
13 13 008B 167 30$: MOVL R7, R10
57 OC C2 008D 168 40$: CMPL R7, R8
56 57 DO 0090 169 : BEQL 50$
66 D5 0093 170 : SUBL #PTR_C_LENGTH, R7
F1 13 0095 171 : MOVL R7, R8
5A OC C2 0097 172 : TSTL (R6)
66 OC 28 009A 173 : BEQL 40$
E8 11 009E 174 : SUBL #PTR_C_LENGTH, R10
5E 59 D1 00A0 175 : MOVC3 #PTR_C_LENGTH, (R6), (R10)
OC 13 00A3 176 : BRB 40$
5A OC C2 00A5 177 50$: CMPL R9, SP
59 OC C2 00A8 178 : BEQL 60$
179 : SUBL #PTR_C_LENGTH, R10
180 : SUBL #PTR_C_LENGTH, R9

: SAVE R10
: GET ADDRESS OF RESULT DESCRIPTOR AR
: COPY ADDRESS OF RESULT DESCRIPTOR A
: ALLOCATE TEMPORARY DESCRIPTOR ARRAY
: SAVE ADDRESS OF TEMPORARY DESCRIPTO
: END OF RESULT DESCRIPTOR ARRAY?
: IF EQL YES
: MOVE DESCRIPTOR TO TEMPORARY ARRAY
: POINT PAST IT IN ORIGINAL ARRAY
: COMMAND QUALIFIER?
: BRANCH IF NOT
: CLEAR COMMAND QUALIFIER DESCRIPTOR
: IN ORIGINAL ARRAY
: POINT TO NEXT ITEM IN TEMPORARY ARR
: QUALIFIER VALUE?
: BRANCH IF NOT
: MOVE DESCRIPTOR TO TEMPORARY ARRAY
: POINT PAST IT IN TEMPORARY ARRAY
: CLEAR COMMAND QUALIFIER DESCRIPTOR
: POINT TO NEXT ITEM IN ORIGINAL ARRA
:
: COPY ADDRESS OF LAST DESCRIPTOR + 1
: ANY MORE DESCRIPTORS TO EXAMINE?
: IF EQL NO
: REMOVE DESCRIPTOR
: GET DESCRIPTOR ADDRESS
: WAS THERE A DESCRIPTOR THERE?
: BRANCH IF NOT
: POINT BEFORE IT IN ORIGINAL ARRAY
: MOVE DESCRIPTOR TO NEW POSITION
:
: ANY MORE TEMPORARY DESCRIPTORS?
: IF EQL NO
: POINT BEFORE IT IN ORIGINAL ARRAY
: POINT BEFORE IT IN TEMPORARY ARRAY
```

6A	69	OC	28	00AB	181	MOV C3	#PTR_C_LENGTH, (R9), (R10)	; MOVE DESCRIPTOR TO NEW POSITION
		EF	11	00AF	182	BRB	50\$	;
				00B1	183			;
5E	0600	CE	9E	00B1	184	MOV AB	WRK_C_RSLBUFSIZ(SP), SP	; DEALLOCATE TEMPORARY DESCRIPTOR ARR
		5A	8E D0	00B6	185	POPL	R10	; RESTORE R10
			05	00B9	186	RSB		;
				00BA	187			
				00BA	188	.END		

DESCRVAL
Symbol table

- TOKEN TABLE MANIPULATION ROUTINES J 13

15-SEP-1984 23:43:53 VAX/VMS Macro V04-00
4-SEP-1984 23:40:15 [DCL.SRC]DESCRVAL.MAR;1Page 6
(4)

DCL\$GETDVAL	00000000	RG	02	WRK_L_TAB_VEC	FFFFFFDE
DCL\$GETNVAL	00000032	RG	02	WRK_L_VERB	FFFFFFBE
DCL\$SORT_TOKENS	0000003C	RG	02	WRK_W_FLAGS	FFFFFFF0
PTR_B_LEVEL	00000004			WRK_W_FLAGS2	FFFFFFF2
PTR_B_NUMBER	00000005			WRK_W_IMGCHAN	FFFFFFEE
PTR_B_PARMCNT	00000006			WRK_W_PMPTLEN	FFFFFF9E
PTR_B_VALUE	00000000				
PTR_C_LENGTH	0000000C				
PTR_K_CMDQUAL	= 00000000				
PTR_K_LENGTH	0000000C				
PTR_K_QUALVALU	= 00000002				
PTR_L_DESCR	00000000				
PTR_L_ENTITY	00000008				
PTR_S_FLAGS	= 00000004				
PTR_S_OFFSET	= 0000000C				
PTR_S_TERM	= 00000004				
PTR_S_TYPE	= 00000004				
PTR_S_VALUE	= 00000008				
PTR_V_FLAGS	= 00000014				
PTR_V_OFFSET	= 00000008				
PTR_V_TERM	= 00000018				
PTR_V_TYPE	= 0000001C				
PTR_V_VALUE	= 00000000				
WRK_B_CMDOPT	FFFFFFC3				
WRK_B_MAXPARM	FFFFFFD0				
WRK_B_MINPARM	FFFFFFD1				
WRK_B_PARMCNT	FFFFFFCE				
WRK_B_PARMSUM	FFFFFFCF				
WRK_B_RECALLCNT	FFFFFFC5				
WRK_B_VALLEV	FFFFFFC4				
WRK_B_VERBTYP	FFFFFFC2				
WRK_C_LENGTH	FFFFF486				
WRK_C_RSLBUFSIZ	= 00000600				
WRK_G_BUFFER	FFFFF492				
WRK_G_INPBUF	FFFFF896				
WRK_G_RESULT	FFFFF9B6				
WRK_K_LENGTH	FFFFF486				
WRK_L_CHARPTR	FFFFF48E				
WRK_L_DISALLOW	FFFFFFE6				
WRK_L_ERRORRTN	FFFFF9AE				
WRK_L_EXPANDPTR	FFFFF486				
WRK_L_IMAGE	FFFFFFE2				
WRK_L_MARKPTR	FFFFF48A				
WRK_L_PAROUT	FFFFFDD2				
WRK_L_PHPTADDR	FFFFF9A2				
WRK_L_PROMPTRTN	FFFFF9A6				
WRK_L_PROPTR	FFFFFFC6				
WRK_L_QUABLK	FFFFFFCA				
WRK_L_READRTN	FFFFF9AA				
WRK_L_RECALLPTR	FFFFFFEA				
WRK_L_RSLEND	FFFFFFB6				
WRK_L_RSLNXT	FFFFFFBA				
WRK_L_SAVAP	FFFFFFF8				
WRK_L_SAVFP	FFFFFFFC				
WRK_L_SAVSP	FFFFFFF4				
WRK_L_SIGNALRTN	FFFFFDD6				
WRK_L_SPECRTN	FFFFF9B2				

+-----+
! 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	FFFFFFFFC (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					
DCL\$ZCODE	000000BA (186.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE					

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	10	00:00:00.09	00:00:01.44
Command processing	84	00:00:00.61	00:00:07.30
Pass 1	129	00:00:02.26	00:00:07.21
Symbol table sort	0	00:00:00.14	00:00:01.04
Pass 2	38	00:00:00.53	00:00:03.21
Symbol table output	6	00:00:00.06	00:00:00.06
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	269	00:00:03.72	00:00:20.28

The working set limit was 900 pages.
9561 bytes (19 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 122 non-local and 7 local symbols.
188 source lines were read in Pass 1, producing 13 object records in Pass 2.
16 pages of virtual memory were used to define 10 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
_\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	2
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	5

207 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:DESCRVAL/OBJ=OBJ\$:DESCRVAL MSRC\$:DESCRVAL/UPDATE=(ENH\$:DESCRVAL)+EXECMLS/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSBLDMLB/LIB

0069 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INTDEF
SOL

COMMAND
LIS

DESCRVAL
LIS

CLIMAC
MAR

CLIMSG
LIS

CONNECT
LIS

DCLPARSE
LIS

CHARMANIP
LIS

EXAMDEP
LIS

EXIT
LIS

INTIMAGES
MAR

DCXSTART
LIS

CLIGBL
LIS

DISALLOW
LIS

CLINT
LIS

CANCEL
LIS