

DDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDD		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
DDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

I I I I I	NN	NN	Q Q Q Q Q	UU	UU	I I I I I	RRRRRRR	EEEEEEEEEE	
I I I I I	NN	NN	Q Q Q Q Q	UU	UU	I I I I I	RRRRRRR	EEEEEEEEEE	
I I	NN	NN	Q Q	UU	UU	I I	RR	RR	EE
I I	NN	NN	Q Q	UU	UU	I I	RR	RR	EE
I I	NNNN	NN	Q Q	UU	UU	I I	RR	RR	EE
I I	NNNN	NN	Q Q	UU	UU	I I	RR	RR	EE
I I	NN NN	NN	Q Q	UU	UU	I I	RRRRRRR	EEEEEEEE	
I I	NN NN	NN	Q Q	UU	UU	I I	RRRRRRR	EEEEEEEE	
I I	NN NN	NN	Q Q	UU	UU	I I	RR RR	EE	
I I	NN NN	NN	Q Q	UU	UU	I I	RR RR	EE	
I I	NN NN	NN	Q Q	UU	UU	I I	RR RR	EE	
I I	NN NN	NN	Q Q	UU	UU	I I	RR RR	EE	
I I	NN NN	NN	Q Q	UU	UU	I I	RR RR	EE	
I I	NN NN	NN	Q Q	UU	UU	I I	RR RR	EE	
I I I I I	NN	NN	Q Q Q Q	UUUUUUUU	UUUUUUUU	I I I I I	RR	RR	EEEEEEEEEE
I I I I I	NN	NN	Q Q Q Q	UUUUUUUU	UUUUUUUU	I I I I I	RR	RR	EEEEEEEEEE

LL	I I I I I	SSSSSSSS
LL	I I I I I	SSSSSSSS
LL	I I	SS
LL	I I	SS
LL	I I	SS
LL	I I	SS
LL	I I	SSSSSS
LL	I I	SSSSSS
LL	I I	SS
LL	I I	SS
LL	I I	SS
LL	I I	SS
LL	I I	SS
LLLLLLLLLL	I I I I I	SSSSSSSS
LLLLLLLLLL	I I I I I	SSSSSSSS

INQUIRE
Table of contents

- INQUIRE DCLS COMMAND EXECUTION C 3

15-SEP-1984 23:58:52 VAX/VMS Macro V04-00

Page 0

(2) 68

INQUIRE COMMAND

```
0000 1 .TITLE INQUIRE - INQUIRE DCLS COMMAND EXECUTION
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 W. H. BROWN 28-JUN-1977
0000 29
0000 30 INQUIRE DCLS COMMAND EXECUTION
0000 31
0000 32 MODIFIED BY:
0000 33
0000 34 V03-007 PCG0008 Peter George 18-Jan-1984
0000 35 Preserve R5 across calls to input routines.
0000 36
0000 37 V03-006 PCG0007 Peter George 12-Jul-1983
0000 38 Fix bug in prompt string parsing.
0000 39 Validate symbol length.
0000 40
0000 41 V03-005 PCG0006 Peter George 27-May-1983
0000 42 Tighten symbol name checking.
0000 43
0000 44 V03-004 PCG0005 Peter George 13-Apr-1983
0000 45 Insert CR/LF before the prompt string.
0000 46
0000 47 V03-003 PCG0004 Peter George 17-Feb-1983
0000 48 Remove reference to $CLIDEFQUALINQU.
0000 49 Use PTR_B_NUMBER to get qualifier numbers.
0000 50
0000 51 V03-002 PCG0003 Peter George 03-Dec-1982
0000 52 Correct bug chopping off first character of the prompt.
0000 53
0000 54 V03-001 PCG0002 Peter George 23-Nov-1982
0000 55 Use prompt descriptor instead of WRK_L_PROMPT.
0000 56 ---
0000 57
```

INQUIRE
V04-000

- INQUIRE DCLS COMMAND EXECUTION E 3

15-SEP-1984 23:58:52 VAX/VMS Macro V04-00
4-SEP-1984 23:41:23 [DCL.SRC]INQUIRE.MAR;1

Page 2
(1)

0000	58	:		
0000	59	:	MACRO LIBRARY CALLS	
0000	60	:		
0000	61	:		
0000	62	:	PRCDEF	: DEFINE PROCESS WORK AREA
0000	63	:	WRKDEF	: DEFINE COMMAND WORK AREA
0000	64	:	PTRDEF	: DEFINE RESULT PARSE DESCRIPTOR
0000	65	:	SYMDEF	: DEFINE TYPES OF SYMBOLS
0000	66	:	\$CLIMSGDEF	: DEFINE ERROR/STATUS CODES

```

0000 68 .SBTTL INQUIRE COMMAND
0000 69 :+
0000 70 : DCL$INQUIRE - INQUIRE COMMAND
0000 71 :
0000 72 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE INQUIRE
0000 73 : DCLS COMMAND.
0000 74 :
0000 75 : INPUTS:
0000 76 :
0000 77 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0000 78 : R9 = ADDRESS OF SCRATCH STACK.
0000 79 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
0000 80 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
0000 81 :
0000 82 : OUTPUTS:
0000 83 :
0000 84 :
0000 85 : R0 LOW BIT CLEAR INDICATES COMMAND FAILURE
0000 86 :
0000 87 : R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
0000 88 :
0000 89 : R0 = DCL$_NORMAL - NORMAL COMPLETION.
0000 90 :-
0000 91 :-
0000 92 .PSECT DCL$ZCODE, BYTE, RD, NOWRT
0000 93
00 2F 20 0000 94 TERMS: .ASCII \ / \<0> ; VALID TERMINATORS-BLANK, SLASH, EOL
20 3A 00 0003 95 PUNC: .ASCII /: / ; TRAILING PUNCTUATION
02 0003
0006 96
0006 97 DCL$INQUIRE::
54 FA AF 9E 0006 98 MOVAB PUNC, R4 ; INQUIRE COMMAND
55 38 AB 9E 000A 99 MOVAB PRC Q_LOCAL(R11), R5 ; ASSUME TO BE PUNCTUATED
00EB 30 000E 100 BSBW 100$ ; SET DEFAULT TABLE ADDRESS
50 20 91 0011 101 CMPB #^A/ /, R0 ; HANDLE QUALIFIERS IF PRESENT
0D 13 0014 102 BEQL 20$ ; TERMINATE WITH A SPACE?
00DB 31 0016 103 8$: BRW 80$ ; IF EQL YES
0019 104 9$: STATUS SYMLNG ; ELSE REPORT ERROR
00DB 31 0020 105 10$: BRW 90$ ; ASSUME SYMBOL IS TOO LONG
FFDA 30 0023 106 20$: BSBW DCL$MARK ; MARK CURRENT PARSE POSITION
FFD7 30 0026 107 BSBW DCL$MOVTKN ; GET THE SYMBOL NAME TOKEN INTO BUFFER
EB 13 0029 108 BEQL 8$ ; BR IF NO SYMBOL NAME
00000FF 8F 51 D1 002B 109 CMPL R1, #255 ; SYMBOL NAME TOO LARGE?
E5 1A 0032 110 BGTRU 9$ ; IF GTRU YES
C4 Af 57 51 7D 0034 111 MOVQ R1, R7 ; SAVE DESCRIPTOR FOR SYMBOL
03 50 3A 0037 112 LOCC R0, #3, TERMS ; CHECK FOR VALID TERMINATOR
D8 13 003C 113 BEQL 8$ ; BR IF INVALID TERMINATOR
FFBF 30 003E 114 BSBW DCL$CHKALPHA ; VERIFY THAT IT IS ACCEPTABLE
DC 50 E9 0041 115 BLBC R0, 10$ ; BR IF NO-REPORT ERROR
00B5 30 0044 116 BSBW 100$ ; ATTEMPT TO CHANGE THE TABLE ASSIGNMENT
B4 Af 03 50 3A 0047 117 LOCC R0, #3, TERMS ; CHECK FOR VALID TERMINATOR
C8 13 004C 118 BEQL 8$ ; BR IF NO-SYNTAX ERROR
FFAF 30 004E 119 BSBW DCL$MOVCHAR ; MOVE DELIMITER (SPACE OR EOL)
0B 12 0051 120 BNEQ 30$ ; BRANCH IF PROMPT SPECIFIED
51 57 7D 0053 121 MOVQ R7, R1 ; NO PROMPT-GET DESCRIPTOR FOR SYMBOL
51 D6 0056 122 INCL R1 ; ADD 1 FOR TERMINATOR
54 D5 0058 123 TSTL R4 ; PUNCTUATION REQUIRED?

```

		38	13	005A	124		BEQL	57\$: IF EQL NO
		19	11	005C	125		BRB	50\$: IF NEQ YES
				005E	126					
		FF9F'	30	005E	127	30\$:	BSBW	DCL\$MARK		: MARK CURRENT PARSE POSITION
		FF9C'	30	0061	128	40\$:	BSBW	DCL\$MOVCHAR		: COPY A BYTE INTO THE EXPANSION BUFFER
F8	F0	AA	04	E0	0064		BBS	#WRK_V_QUOTE,WRK_W_FLAGS(R10),40\$: WHILE WITHIN QUOTES
		20	50	91	0069		CMPB	RO,#*X20		: BLANK DETECTED OUTSIDE QUOTES?
			AB	13	006C		BEQL	8\$: YES, THEN SYNTAX ERROR
			50	D5	006E		TSTL	RO		: END OF LINE?
			EF	12	0070		BNEQ	40\$: NO, CONTINUE
			FF8B'	30	0072		BSBW	DCL\$MARKEDTOKEN		: GET DESCRIPTOR OF STRING
			51	D7	0075		DECL	R1		: DISCOUNT EOL CHARACTER
					0077					
			54	DD	0077	50\$:	PUSHL	R4		: SAVE PUNCTUATION INDICATOR
			FF84'	30	0079		BSBW	DCL\$COMPRESS		: COMPRESS THE STRING
			54	BED0	007C		POPL	R4		: RETRIEVE PUNC. INDICATOR - ANY REQUIRED?
			13	13	007F		BEQL	57\$: IF EQL, NO
			51	84	0081		ADDB	(R4)+,R1		: ADD LENGTH OF PUNCTUATION
50	F486	CA	D0	0084	142		MOVL	WRK_L_EXPANDPTR(R10),RO		: GET EXPANSION BUFFER POINTER
			80	84	0089		MOVB	(R4)+,(RO)+		: MOVE FIRST BYTE TO EOL MARKER
			80	64	008C		MOVZBL	(R4),(RO)+		: MOVE IN 4 MORE BYTES OF IT
	F486	CA	05	C0	008F		ADDL	#5,WRK_L_EXPANDPTR(R10)		: INCREMENT EXPANSION POINTER
					0094					
000001FE	BF	51	D1	0094	147	57\$:	CMPL	R1,#WRK_C_SCRSTKSIZ-2		: WILL STRING FIT IN PROMPT BUFFER?
			4E	1A	009B		BGTRU	75\$: NO, THEN SIGNAL ERROR
	F99E	CA	51	B0	009D		MOVW	R1,WRK_W_PMPTLEN(R10)		: SET PROMPT STRING LENGTH
	59	FE00	C9	9E	00A2		MOVAB	-WRK_C_SCRSTKSIZ(R9),R9		: GET BASE OF SCRATCH STACK
	F9A2	CA	59	D0	00A7		MOVL	R9,WRK_L_PMPTADDR(R10)		: SET PROMPT STRING POINTER
	69	0A0D	8F	B0	00AC		MOVW	#*X0A0D,(R9)		: MOVE CR/LF ONTO THE SCRATCH STACK
	F99E	CA	02	A0	00B1		ADDW	#2,WRK_W_PMPTLEN(R10)		: ADD IN LENGTH OF CR/LF
			55	DD	00B6		PUSHL	R5		: SAVE SYMBOL TABLE ADDRESS
02	A9	62	51	28	00B8		MOVW3	R1,(R2),2(R9)		: MOVE PROMPT ONTO THE SCRATCH STACK
					00BD		SETBIT	WRK_V_INQUIRE,WRK_W_FLAGS(R10)		: INDICATE DOING A INQUIRE
					00C2		SETBIT	PRC_V_IND,PRC_W_FLAGS(R11)		: DISABLE INDIRECT RECOGNITION
			FF37'	30	00C6		BSBW	DCL\$FORNBLK		: MAKE AN INPUT HAPPEN
			FF34'	30	00C9		BSBW	DCL\$MARK		: MARK CURRENT POSITION IN BUFFER
			FF31'	30	00CC	60\$:	BSBW	DCL\$MOVCHAR		: GET THE ANSWER
			FB	12	00CF		BNEQ	60\$: UNTIL END OF LINE
			FF2C'	30	00D1		BSBW	DCL\$MARKEDTOKEN		: GET DESCRIPTOR OF STRING
			51	D7	00D4		DECL	R1		: REMOVE COUNT FOR EOL
			03	13	00D6		BEQL	70\$: BR IF NULL STRING IS THE ANSWER
			FF25'	30	00D8		BSBW	DCL\$COMPRESS		: COMPRESS QUOTES
			53	57	7D	70\$:	MOVQ	R7,R3		: SET DESCRIPTOR FOR SYMBOL
			50	00	D0		MOVL	#SYM_K_STRING,RO		: SET TYPE IS STRING
			55	BED0	00E1		POPL	R5		: RESTORE SYMBOL TABLE ADDRESS
68	AB	04	AB	00E4	169		BISW	#PRC_M_DISABL,PRC_W_FLAGS(R11)		: DISABLE CONTROL Y/C AST'S
			FF15'	31	00E8		BRW	DCL\$ALCOCSYM		: CREATE THE SYMBOL DEFINITION
					00EB					
					00EB					
					00EB					
					00EB	75\$:	STATUS	BUFOVF		: BUFFER OVERFLOW
			07	11	00F2		BRB	90\$: ERROR STATUS
					00F4	80\$:	STATUS	PARMDEL		: RETURN WITH ERROR
				05	00FB	90\$:	RSB			
					00FC					
					00FC					
					00FC					
					180					: LOCAL SUBROUTINE TO PROCESS QUALIFIERS

				00FC	181	:				
				00FC	182	:				
				00FC	183	:				
				00FC	184	:				
				00FC	185	:				
				00FC	186	:				
				00FC	187	:				
				00FC	188	:				
				00FC	189	:				
				00FC	190	100\$:	BSBW	DCL\$SETCHAR	: PEEK AT NEXT CHARACTER IN INPUT	
				00FF	191		BSBW	DCL\$MARK	: MARK CURRENT PARSE POSITION	
	2F	50	91	0102	192		CMPB	RO,#^A\^A	: QUALIFIER?	
		45	12	0105	193		BNEQ	120\$: BR IF NO QUALIFIER IN SIGHT	
	0230	8F	BB	0107	194		PUSHR	#^M<R4,R5,R9>	: SAVE INDICATORS	
			30	010B	195		BSBW	DCL\$PROCQUAL	: PROCESS THE QUALIFIER	
	0230	8F	BA	010E	196		POPR	#^M<R4,R5,R9>	: RETRIEVE INDICATORS	
		35	50	E9	197		BLBC	RO,110\$: BR IF ERROR IN QUALIFIER	
51	BA	AA	0C	C3	0115	198	SUBL3	#PTR_C_LENGTH,WRK_L_RSLNXT(R10),R1	: ADDRESS OF PREVIOUS TOKEN DESCR	
	00'8F		05	A1	91	011A	199	CMPB	PTR_B_NUMBER(R1),#CCISK_INQU_PUNC	: PUNCTUATION QUALIFIER?
				0F	12	011F	200	BNEQ	105\$: IF NEQ NO
	04	61		E1	0121	201	B9C	#PTR_V_NEGATE,(R1),102\$: BRANCH IF NOT NEGATED?	
				D4	0125	202	CLRL	R4	: INDICATE NO PUNCTUATION	
				D3	0127	203	BRB	100\$		
					0129	204			102\$:	
	54	FED6	CF	9E	0129	205	MOVAB	PUNC,R4	: SET PUNC STRING ADDRESS	
			CC	11	012E	206	BRB	100\$		
					0130	207			105\$:	
	00'8F		05	A1	91	0130	208	CMPB	PTR_B_NUMBER(R1),#CLISK_INQU_GLOB	: GLOBAL QUALIFIER?
				06	12	0135	209	BNEQ	107\$: IF NEQ NO
	55	28	AB	9E	0137	210	MOVAB	PRC_Q_GLOBAL(R11),R5	: SET PROPER TABLE ADDRESS	
				BF	11	013B	211	BRB	100\$: TRY FOR MORE
					013D	212			107\$:	
	00'8F		05	A1	91	013D	213	CMPB	PTR_B_NUMBER(R1),#CLISK_INQU_LOCA	: LOCAL QUALIFIER?
				B8	12	0142	214	BNEQ	100\$: IF NEQ NO, IGNORE IT
	55	38	AB	9E	0144	215	MOVAB	PRC_Q_LOCAL(R11),R5	: SET LOCAL SYMBOL TABLE	
				B2	11	0148	216	BRB	100\$	
				8E	D5	014A	217	TSTL	(SP)+	: CLEAR RETURN ADDRESS
				05	014C	218	RSB		: RETURN TO CALLER	
					014D	219				
					014D	220			.END	

OUTPUTS:

R0 = TERMINATING CHARACTER
R4 = 0 IF /NOPUNCTUATION
ADDRESS OF PUNCTUATION STRING IF /PUNCT
R5 = ADDRESS OF GLOBAL SYMBOL TABLE DESCRIPTOR
IF /GLOBAL SPECIFIED.
ADDRESS OF LOCAL SYMBOL TABLE IF /LOCAL

INQUIRE
Symbol table

- INQUIRE DCLS COMMAND EXECUTION

I 3

15-SEP-1984 23:58:52 VAX/VMS Macro V04-00
4-SEP-1984 23:41:23 [DCL.SRC]INQUIRE.MAR;1

CLISK_INQU_GLOB	*****	X	02	PRC_L_SEVERITY	00000050		
CLISK_INQU_LOCA	*****	X	02	PRC_L_SPWN	000000C0		
CLISK_INQU_PUNC	*****	X	02	PRC_L_STACKLM	000000A4		
CLIS_BUFOVF	= 00038018			PRC_L_STACKPT	000000A0		
CLIS_PARMDEL	= 00038110			PRC_L_STATUS	00000054		
CLIS_SYMLNG	= 00038270			PRC_L_STS	00000084		
DCL\$ALLOCSYM	*****	X	02	PRC_L_STV	00000088		
DCL\$CHKALPHA	*****	X	02	PRC_L_SYMBOL	00000060		
DCL\$COMPRESS	*****	X	02	PRC_L_TMBX	00000074		
DCL\$FORNBLK	*****	X	02	PRC_L_TRMLIST	00000010		
DCL\$INQUIRE	00000006	RG	02	PRC_M_DISABL	= 00000004		
DCL\$MARK	*****	X	02	PRC_Q_ALLOCREG	00000020		
DCL\$MARKEDTOKEN	*****	X	02	PRC_Q_COMMAND	000000E0		
DCL\$MOVCHAR	*****	X	02	PRC_Q_FLUSHTIME	000000D0		
DCL\$MOVTOKN	*****	X	02	PRC_Q_GLOBAL	00000028		
DCL\$PROCQUAL	*****	X	02	PRC_Q_IMAGENAME	000000D8		
DCL\$SETCHAR	*****	X	02	PRC_Q_KEYPAD	00000040		
PRC_B_CONTINUE	000000F3			PRC_Q_LABEL	00000030		
PRC_B_DEFRADIX	000000AE			PRC_Q_LOCAL	00000038		
PRC_B_EXMDEPMOD	000000AD			PRC_Q_SAVEPRIV	000000E8		
PRC_B_EXMDEPWID	000000AC			PRC_T_OUTDVI	0000011C		
PRC_B_EXONLYL	0000012D			PRC_V_IND	= 00000005		
PRC_B_FLAGS2	000000AF			PRC_W_ASTIOSB	000000C6		
PRC_B_IMGFLAG	00000078			PRC_W_ASTRETN	000000C8		
PRC_B_OUTFLAGS	0000012C			PRC_W_ASTSTATUS	000000C4		
PRC_B_PROMPTLEN	000000F0			PRC_W_ATTMBX	0000007A		
PRC_C_LENGTH	00000534			PRC_W_FLAGS	00000068		
PRC_G_COMMANDS	00000133			PRC_W_INPCHAN	00000064		
PRC_G_PROMPT	000000F4			PRC_W_ONLEVEL	0000006A		
PRC_K_LENGTH	00000534			PRC_W_OUTIFI	00000114		
PRC_L_CURRKEY	00000048			PRC_W_OUTISI	00000116		
PRC_L_EXMDEPADR	000000A8			PRC_W_OUTMBXCHN	000000CA		
PRC_L_EXTARG	00000094			PRC_W_OUTMBXREF	000000CE		
PRC_L_EXTBLK	0000008C			PRC_W_OUTMBXSIZ	000000CC		
PRC_L_EXTCOD	0000009C			PRC_W_PMPTCTRL	000000F1		
PRC_L_EXTHND	00000090			PRC_W_WAITIOSB	00000066		
PRC_L_EXTPRM	00000098			PTR_B_LEVEL	00000004		
PRC_L_IDFLNK	0000008C			PTR_B_NUMBER	00000005		
PRC_L_IMGACTSTS	00000080			PTR_B_PARMCNT	00000006		
PRC_L_INDCLOCK	0000007C			PTR_B_VALUE	00000000		
PRC_L_INDEPTH	0000005C			PTR_C_LENGTH	0000000C		
PRC_L_INDFAB	0000001C			PTR_K_LENGTH	0000000C		
PRC_L_INDIRPRAB	00000014			PTR_L_DESCR	00000000		
PRC_L_INDOUTRAB	00000018			PTR_L_ENTITY	00000008		
PRC_L_INPRAB	00000008			PTR_V_NEGATE	= 00000014		
PRC_L_LASTKEY	0000004C			PUNC	00000003	R	02
PRC_L_LSTSTATUS	00000080			SYM_B_FLAGS	0000000B		
PRC_L_ONCTLY	00000088			SYM_B_NONUNIQUE	0000000B		
PRC_L_ONERROR	0000006C			SYM_B_TYPE	0000000A		
PRC_L_OUTOFBAND	000000B4			SYM_K_STRING	= 00000000		
PRC_L_OUTRAB	0000000C			SYM_L_BL	00000004		
PRC_L_OUTRABCTX	00000118			SYM_L_FL	00000000		
PRC_L_PPFLIST	00000070			SYM_T_SYMBOL	0000000C		
PRC_L_RECALLPTR	0000012F			SYM_W_SIZE	00000008		
PRC_L_RESTART	000G0058			TERMS	00000000	R	02
PRC_L_SAVAP	00000000			WRK_B_CMDOPT	FFFFFFFFC3		
PRC_L_SAVFP	00000004			WRK_B_MAXPARM	FFFFFFFFD0		

INQUIRE
Symbol table

- INQUIRE DCLS COMMAND EXECUTION

J 3

15-SEP-1984 23:58:52 VAX/VMS Macro V04-00
4-SEP-1984 23:41:23 [DCL.SRC]INQUIRE.MAR;1

Page 7
(2)

```

WRK_B_MINPARM          FFFFFFFD1
WRK_B_PARMCNT          FFFFFFFCE
WRK_B_PARMSUM          FFFFFFFCF
WRK_B_RECALLCNT        FFFFFFFC5
WRK_B_VALLEV           FFFFFFFC4
WRK_B_VERBTYP          FFFFFFFC2
WRK_C_LENGTH           FFFFF486
WRK_C_SCRSTKSIZ        = 00000200
WRK_G_BUFFER           FFFFF492
WRK_G_INPBUF           FFFFF896
WRK_G_RESULT           FFFFF9B6
WRK_K_LENGTH           FFFFF486
WRK_L_CHARPTR          FFFFF48E
WRK_L_DISALLOW         FFFFFE6
WRK_L_ERRORRTN         FFFFF9AE
WRK_L_EXPANDPTR        FFFFF486
WRK_L_IMAGE            FFFFFE2
WRK_L_MARKPTR          FFFFF48A
WRK_L_PAROUT           FFFFFFFD2
WRK_L_PMPTADDR         FFFFF9A2
WRK_L_PROMPTRTN        FFFFF9A6
WRK_L_PROPTR           FFFFFFFC6
WRK_L_QUABLK           FFFFFFFCA
WRK_L_READRTN          FFFFF9AA
WRK_L_RECALLPTR        FFFFFFFEA
WRK_L_RSLND            FFFFFFFB6
WRK_L_RSLNXT           FFFFFFFBA
WRK_L_SAVAP            FFFFFFFF8
WRK_L_SAVFP            FFFFFFFFC
WRK_L_SAVSP            FFFFFFFF4
WRK_L_SIGNALRTN        FFFFFFFD6
WRK_L_SPECTRN          FFFFF9B2
WRK_L_TAB_VEC          FFFFFFFDE
WRK_L_VERB             FFFFFFFBE
WRK_V_INQUIRE         = 00000007
WRK_V_QUOTE            = 00000004
WRK_W_FLAGS            FFFFFFFF0
WRK_W_FLAGS2           FFFFFFFF2
WRK_W_IMGCHAN          FFFFFFFEE
WRK_W_PMPTLEN          FFFFF99E
_SS_                   = 000000EF

```

↑-----↑
! 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	FFFFFFFC (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
DCL\$ZCODE	0000014D (333.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	10	00:00:00.05	00:00:00.95
Command processing	82	00:00:00.71	00:00:06.09
Pass 1	193	00:00:06.24	00:00:22.88
Symbol table sort	0	00:00:00.64	00:00:02.20
Pass 2	45	00:00:01.02	00:00:03.83
Symbol table output	18	00:00:00.13	00:00:00.13
Psect synopsis output	1	00:00:00.03	00:00:00.05
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	349	00:00:08.83	00:00:36.14

The working set limit was 1050 pages.
27034 bytes (53 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 464 non-local and 19 local symbols.
220 source lines were read in Pass 1, producing 14 object records in Pass 2.
33 pages of virtual memory were used to define 18 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
_\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	8
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	11

592 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:INQUIRE/OBJ=OBJ\$:INQUIRE MSRC\$:INQUIRE/UPDATE=(ENHS\$:INQUIRE)+EXECMLS\$/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSBLDMLB/LIB

0071 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INQUIRE
LIS

LEXICON
LIS

KEYPAD
LIS

LOGICAL
LIS