

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

```
GGGGGGGG 000000 TTTTTTTTTT 000000
GGGGGGGG 000000 TTTTTTTTTT 000000
GG        00        00        00        00
GG        00        00        00        00
GG        00        00        00        00
GG        00        00        00        00
GG        00        00        00        00
GG  GGGGGG 00        00        00        00
GG  GGGGGG 00        00        00        00
GG    GG    00        00        00        00
GG    GG    00        00        00        00
GGGGGG 000000 TTT        000000
GGGGGG 000000 TTT        000000
```

....
....
....
....

```
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
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS
```

GOTO
Table of contents

- GOTO COMMAND EXECUTION

G 7

15-SEP-1984 23:49:46 VAX/VMS Macro V04-00

Page 0

(3) 75
(4) 248
(5) 319
(6) 352

GOTO LABEL
FLUSH CURRENT RECORD AND ANY CONTINUATIONS
ALLOCATE A LABEL SYMBOL TABLE ENTRY
DEALLOCATE GOTO CONTROL BLOCK

```

0000 1 .TITLE GOTO - GOTO 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 GOTO COMMAND EXECUTION
0000 29
0000 30 D. N. CUTLER 6-MAY-77
0000 31
0000 32 MODIFIED BY:
0000 33
0000 34 V03-007 HWS0003 Harold Schultz 06-FEB-1984
0000 35 Treat lines that begin without a '$' and end with a
0000 36 '-' as a line of data and not as a command continuation.
0000 37
0000 38 V03-006 PCG0006 Peter George 28-Nov-1983
0000 39 Allow blanks to appear between label and trailing colon.
0000 40
0000 41 V03-005 PCG0005 Peter George 15-Jul-1983
0000 42 Use multi-national upcase algorithm.
0000 43 Support null nodes.
0000 44
0000 45 V03-004 PCG0004 Peter George 23-Jun-1983
0000 46 Clear DCL_L_SYMBOL after deallocating it.
0000 47
0000 48 V03-003 PCG0003 Peter George 25-Apr-1983
0000 49 Do not return if success. Branch to DCL$RESTART.
0000 50
0000 51 V03-002 PCG0002 Peter George 01-Apr-1983
0000 52 Perform forward goto scanning here.
0000 53
0000 54 V03-001 PCG0001 Peter George 24-feb-1983
0000 55 Remove SETBIT WRK_V_NOSTAT.
0000 56 :---

```

```
0000 58 :  
0000 59 : MACRO LIBRARY CALLS  
0000 60 :  
0000 61 : PRCDEF ;DEFINE PROCESS WORK AREA  
0000 62 : WRKDEF ;DEFINE COMMAND WORK AREA  
0000 63 : SYMDEF ;DEFINE SYMBOL ENTRY OFFSETS  
0000 64 : $CLMSGDEF ;DEFINE ERROR/STATUS VALUES  
0000 65 : $DEVDEF ;DEFINE DEVICE CHARACTERISTIC BITS  
0000 66 : $RABDEF ;DEFINE RAB OFFSETS  
0000 67 : $STSDEF ;DEFINE MESSAGE BITS  
0000 68 :  
00000000 69 : .PSECT DCL$ZCODE, BYTE, RD, NOWRT  
0000 70 :  
SE 3D 3F 3E 5D 29 28 2D 2C 2B 2F 3A 0000 71 TERM: .ASCII \:/+,-(])>?=#@*'[<.: ! \  
09 21 20 3B 2E 3C 5B 27 2A 40 23 000C  
00 0017 72 : .BYTE 0  
0018 73 TERMEND:
```

```

0018 75 .SBTTL GOTO LABEL
0018 76 ;+
0018 77 DCL$GOTO - GOTO LABEL
0018 78
0018 79 THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE GOTO LABEL DCLS
0018 80 COMMAND.
0018 81
0018 82 R10 = BASE ADDRESS OF COMMAND WORK AREA.
0018 83 R11 = BASE ADDRESS OF PROCESS WORK AREA.
0018 84
0018 85 OUTPUTS:
0018 86
0018 87 THE SPECIFIED LABEL IS SEARCHED FOR IN THE CURRENT LABEL TABLE, AND IF
0018 88 FOUND, A RANDOM FIND IS EXECUTED TO POSITION THE INPUT FILE FOR SUBSE-
0018 89 QUENT GET OPERATIONS. OTHERWISE A FORWARD GOTO SEARCH IS EXECUTED.
0018 90
0018 91
0018 92 DCL$GOTO:: ;GOTO LABEL
0018 93
0018 94
0018 95 DISABLE CTRL/Y AST'S.
0018 96
0018 97 DISABLE ;DISABLE CTRL/Y'S
001E 98 CLRQ (SP)+ ;REMOVE RETURN INFO FROM STACK
0020 99 SETBIT WRK_V_NOSTAT,WRK_W_FLAGS(R10) ;DO NOT CHANGE $STATUS ON SUCCESS
0025 100
0025 101
0025 102 CHECK IF GOTO'S ARE ALLOWED.
0025 103
0025 104 MOVL PRC_L_INDINPRAB(R11),R0 ;GET CURRENT INPUT RAB
3B 18 A0 1C E1 0029 105 BBC #DEV$V_RND,RAB$L_CTX(R0),10$ ;IF CLR, NOT RANDOM ACCESS DEVICE
002E 106
002E 107
002E 108 GET DESTINATION. CHECK IF IT'S ALREADY DEFINED.
002E 109
002E 110 BSBW DCL$GETDVAL ;GET LABEL DESCRIPTOR VALUES
0031 111 MOVQ R1,-(SP) ;SAVE LABEL DESCRIPTOR
50 7E 30 AB 9E 0034 112 MOVAB PRC_Q_LABEL(R11),R0 ;GET ADDRESS OF LABEL TABLE LISTHEAD
0038 113 BSBW DCL$SEARCHT ;SEARCH TABLE FOR LABEL
51 8E 7D 003B 114 MOVQ (SP)+,R1 ;RESTORE LABEL DESCRIPTOR
38 50 E9 003E 115 BLBC R0,30$ ;IF LBC LABEL NOT FOUND
0041 116
0041 117
0041 118 LABEL DEFINED ~ POINT TO PROPER RECORD
0041 119
0041 120 MOVL R3,R6 ;COPY SYMBOL ADDRESS
0044 121 MOVQ R1,R4 ;COPY LABEL DESCRIPTOR
10 A7 0F A644 06 28 0047 122 5$: MOVL PRC_L_INDINPRAB(R11),R7 ;GET ADDRESS OF INDIRECT FILE RAB
004B 123 MOVC #6,$SYM T SYMBOL+3(R6)[R4],- ;RESTORE RECORD FILE ADDRESS
0052 124 RAB$W_RFA(R7)
0052 125 MOVB #RAB$C_RFA,RAB$B_RAC(R7) ;SET ACCESS MODE FOR RANDOM FIND
0056 126 $FIND RAB=(R7) ;FIND RECORD THAT HAS LABEL
005F 127 BLBC R0,20$ ;SIGNAL ANY ERRORS
1E A7 0E 50 E9 0062 128 MOVB #RAB$C_SEQ,RAB$B_RAC(R7) ;RESET TO SEQUENTIAL ACCESS MODE
1E A7 00 90 0066 129 BRW DCL$RESTART ;EXECUTE THE SPECIFIED COMMAND
FF97 31 0069 130 10$: STATUS NORMAL ;SET NORMAL STATUS
05 0070 131 20$: RSB ;RETURN

```

```

0071 132
0071 133
0071 134 : RETURN SYMOVF.
0071 135
0071 136 22$: STATUS SYMOVF
05 0078 137 RSB
0079 138
0079 139
0079 140 : LABEL NOT DEFINED. SAVE IT IN THE SYMBOL TABLE.
0079 141
54 51 7D 0079 142 30$: MOVQ R1,R4 ;SAVE DESCRIPTOR OF LABEL
51 01 C0 007C 143 ADDL #1,R1 ;ADD IN LABEL SIZE BYTE
FF7E' 30 007F 144 BSBW DCL$ALLDYNMEM ;ALLOCATE DYNAMIC MEMORY
EC 50 E9 0082 145 BLBC R0,22$ ;BR IF FAILED
60 AB 52 D0 0085 146 MOVL R2,PRC_L_SYMBOL(R11) ;SAVE TARGET FOR GOTO
82 54 90 0089 147 MOVB R4,(R2)+ ;SET SIZE OF LABEL STRING
62 65 54 28 008C 148 MOVC R4,(R5),(R2) ;FILL IN LABEL NAME
0090 149
0090 150
0090 151 : LOOK AHEAD FOR THE LABEL.
0090 152
68 AB 10 A8 0090 153
68 AB 20 A8 0094 154 40$: BISW #PRC_M_GOTO,PRC_W_FLAGS(R11) ;SET FORWARD GOTO IN PROGRESS
FF65' 30 0098 155 BISW #PRC_M_IND,PRC_W_FLAGS(R11) ;IGNORE @ SIGNS
68 AB 20 AA 009B 156 BSBW DCL$INPUT ;GET NEXT COMMAND
25 68 AB 04 E1 009F 157 BICW #PRC_M_IND,PRC_W_FLAGS(R11) ;RESTORE @ SIGNS
53 F48E CA D0 00A4 158 BBC #PRC_V_GOTO,PRC_W_FLAGS(R11),25$ ;BRANCH IF EOF
53 53 D7 00A9 159 MOVL WRK_C_CHARPTR(RT0),R3 ;POINT TO FIRST CHARACTER
00AB 160 DECL R3 ;BACK UP A CHAR
00AB 161
00AB 162 : IGNORE LEADING BLANKS, LEADING '$', AND LINES THAT ARE ENTIRELY BLANK.
00AB 163
50 D4 00AB 164
53 D6 00AD 165 50$: CLRL R0 ;ASSUME NO '$'
63 95 00AF 166 INCL R3 ;GET NEXT CHAR
E1 13 00B1 167 TSTB (R3) ;EOL?
63 20 91 00B3 168 BEQL 40$ ;YES, THEN GET NEXT LINE
F5 13 00B6 169 CMPB #^X20,(R3) ;SKIP LEADING BLANKS
63 09 91 00B8 170 BEQL 50$
F0 13 00BB 171 CMPB #^X09,(R3)
63 24 91 00BD 172 BEQL 50$
0F 12 00C0 173 CMPB #^A/$/, (R3) ;SKIP LEADING '$'
E7 50 00  E3 00C2 174 BNEQ 53$ ;BRANCH IF NOT
0083 31 00C6 175 BBCS #0,R0,50$ ;SET FLAG, GET NEXT CHAR
00C9 176 BRW 70$ ;FLUSH IF PREVIOUSLY SET
00C9 177
00C9 178 : EOF ENCOUNTERED. RETURN USGOTO.
00C9 179
50 10038148 8F D0 00C9 180 25$: MOVL #CLIS_USGOTO!STSM_INHIB_MSG,R0 ;SET UNSATISFIED GOTO STATUS
05 00D0 181 RSB
00D1 182
00D1 183
00D1 184 : DETERMINE IF THERE IS A LABEL ON THIS LINE.
00D1 185
FF26 CF 52 53 D0 00D1 186 53$: MOVL R3,R2 ;SAVE START OF LABEL PTR
18 63 3A 00D4 187 55$: LOCC (R3),#TERMEND-TERM,TERM ;FIND THE FIRST TERMINATOR
1F 12 00DA 188 BNEQ 58$ ;BRANCH IF FOUND

```

```

61 8F 63 91 00DC 189      CMPB (R3),#^A/a/      ;CHECK LOW LIMIT OF LOW RANGE
      15 1F 00E0 190      BLSSU 57$             ;BR IF FAILED
7A 8F 63 91 00E2 191      CMPB (R3),#^A/z/      ;CHECK HIGH LIMIT OF LOW RANGE
      0C 1B 00E6 192      BLEQU 56$             ;BR IF VALID CHARACTER
EO 8F 63 91 00E8 193      CMPB (R3),#^XE0          ;CHECK LOW LIMIT OF HIGH RANGE
      09 1F 00EC 194      BLSSU 57$             ;BR IF FAILED
FE 8F 63 91 00EE 195      CMPB (R3),#^XFE          ;CHECK HIGH LIMIT OF HIGH RANGE
      03 1A 00F2 196      BGTRU 57$             ;BR IF FAILED
      63 20 8A 00F4 197 56$: BICB #^X20,(R3)        ;UPCASE THE CHARACTER
      53 D6 00F7 198 57$: INCL R3                 ;GET NEXT CHAR
      D9 11 00F9 199      BRB 55$                ;CHECK NEXT CHAR
      00FB 200
51 53 D0 00FB 201 58$:  MOVL R3,R1                 ;SAVE END OF LABEL PTR
      02 11 00FE 202      BRB 60$                ;
      0100 203
      53 D6 0100 204 59$:  INCL R3                 ;GET NEXT CHAR
      63 20 91 0102 205 60$: CMPB #^X20,(R3)        ;SKIP BLANKS
      F9 13 0105 206      BEQL 59$                ;
      63 09 91 0107 207      CMPB #^X09,(R3)        ;
      F4 13 010A 208      BEQL 59$                ;
      63 3A 91 010C 209      CMPB #^A/:/, (R3)        ;WAS IT A COLON?
      3B 12 010F 210      BNEQ 70$                ;NO, FLUSH THIS RECORD
01 A3 3A 91 0111 211      CMPB #^A/:/,1(R3)        ;IS IT A NULL NODE?
      35 13 0115 212      BEQL 70$                ;YES, FLUSH THIS RECORD
01 A3 3D 91 0117 213      CMPB #^A/=/,1(R3)        ;IS IT AN ASSIGNMENT?
      2F 13 011B 214      BEQL 70$                ;YES, FLUSH THIS RECORD
      011D 215
      011D 216
      011D 217 : A LABEL HAS BEEN FOUND. INSERT IT IN THE LABEL TABLE.
      011D 218
51 52 C2 011D 219      SUBL R2,R1                 ;GET LENGTH OF LABEL
      009C 30 0120 220      BSBW DCL$ALLOC_LABEL      ;ALLOCATE A LABEL SYMBOL FOR IT
      0A 50 E8 0123 221      BLBS R0,65$           ;BRANCH IF SUCCESS
68 AB 10 AA 0126 222      BICW #PRC M GOTO,PRC_W_FLAGS(R11) ;CLEAR GOTO IN PROGRESS FLAGS
      00B4 30 012A 223      BSBW DCL$DEALGOTO        ;DEALLOCATE THE GOTO SYMBOL
      FF41 31 012D 224      BRW 22$                ;SIGNAL THE ERROR
      0130 225
      0130 226
      0130 227 : DOES IT MATCH THE ONE WE WERE LOOKING FOR? IF SO, RESET RMS CONTEXT
      0130 228 : TO REREAD THIS RECORD.
      0130 229
52 60 AB D0 0130 230 65$: MOVL PRC_L_SYMBOL(R11),R2 ;GET TARGET OF GOTO
      51 82 9A 0134 231      MOVZBL (R2)+,R1        ;CREATE LABEL DESCRIPTOR
      51 54 D1 0137 232      CMPL R4,R1           ;DOES LENGTH MATCH?
      10 12 013A 233      BNEQ 70$                ;NO, FLUSH THIS RECORD
62 65 54 29 013C 234      CMPC3 R4,(R5),(R2)        ;DOES THE STRING MATCH
      0A 12 0140 235      BNEQ 70$                ;NO, FLUSH THIS RECORD
68 AB 10 AA 0142 236      BICW #PRC M GOTO,PRC_W_FLAGS(R11) ;CLEAR GOTO IN PROGRESS FLAGS
      0098 30 0146 237      BSBW DCL$DEALGOTO        ;DEALLOCATE THE GOTO SYMBOL
      FEFB 31 0149 238      BRW 5$                ;POSITION FILE TO GOTO LABEL
      014C 239
      014C 240
      014C 241 : FLUSH THE RECORD.
      014C 242
      014C 243
03 68 AB 0B 10 014C 243 70$: BSBB FLUSH             ;FLUSH THE RECORD
      04 E1 014E 244      BBC #PRC_V_GOTO,PRC_W_FLAGS(R11),71$ ;BRANCH IF EOF
      FF3E 31 0153 245      BRW 40$                ;GET THE NEXT RECORD

```


GOTO
V04-000

- GOTO COMMAND EXECUTION
GOTO LABEL

M 7

15-SEP-1984 23:49:46 VAX/VMS Macro V04-00
4-SEP-1984 23:40:47 [DCL.SRC]GOTO.MAR;1

Page 6
(3)

FF70 31 0156 246 71\$: BRW 25\$

:

```

0159 248 .SBTTL FLUSH CURRENT RECORD AND ANY CONTINUATIONS
0159 249 :+
0159 250 : FLUSH - FLUSH CURRENT RECORD AND ANY CONTINUATIONS
0159 251 :
0159 252 : THIS ROUTINE IS CALLED TO FLUSH CURRENT RECORD AND ANY CONTINUATIONS.
0159 253 :
0159 254 : INPUTS:
0159 255 :
0159 256 : R10 = ADDRESS OF WRK DATA STRUCTURE
0159 257 : R11 = ADDRESS OF PRC DATA STRUCTURE
0159 258 :
0159 259 : OUTPUTS:
0159 260 :
0159 261 : R0 = STATUS
0159 262 :-
0159 263 :
0159 264 FLUSH:
0159 265 CLRL -(SP) ;FLUSH THE INPUT
0159 266 CMPB @WRK_L_CHARPTR(R10),#^A/$/ ;USE STACK AS TEMP. FLAG
0159 267 BNEQ 2$ ;IS FIRST CHAR. A '$' ?
0159 268 INCL (SP) ;NO, TREAT LINE AS DATA.
;YES, TREAT AS A COMMAND LINE
0164 269 :
0164 270 : POINT TO FIRST CHARACTER IN RECORD AND INIT QUOTE FLAG.
0164 271 :
51 F48E CA D0 0164 272 2$: MOVL WRK_L_CHARPTR(R10),R1 ;POINT TO FIRST CHARACTER
51 51 D7 0169 273 DECL R1 ;BACK UP A CHAR
52 51 D0 016B 274 MOVL R1,R2 ;SAVE UNDERFLOW CHAR PTR
53 53 D4 016E 275 CLRL R3 ;INIT QUOTE FLAG
0170 276 :
0170 277 :
0170 278 : SEARCH FOR EOL OR TRAILING COMMENT.
0170 279 :
61 51 D6 0170 280 10$: INCL R1 ;GET NEXT CHAR
61 61 95 0172 281 TSTB (R1) ;EOL?
61 13 13 0174 282 BEQL 40$ ;YES, THEN CHECK FOR "'-'
61 22 91 0176 283 CMPB #^A/'',(R1) ;QUOTE?
61 04 12 0179 284 BNEQ 20$ ;BRANCH IF NOT
53 D6 017B 285 INCL R3 ;TOGGLE THE FLAG
EE F1 11 017D 286 BRB 10$ ;GET NEXT CHAR
61 EE 53 E8 017F 287 20$: BLBS R3,10$ ;BRANCH IF IN QUOTES
61 21 91 0182 288 CMPB #^A/!/, (R1) ;COMMENT?
61 02 13 0185 289 BEQL 40$ ;YES, THEN CHECK FOR "'-'
61 E7 11 0187 290 BRB 10$ ;GET NEXT CHAR
0189 291 :
0189 292 :
0189 293 : BACK UP OVER TRAILING BLANKS.
0189 294 :
52 51 D7 0189 295 40$: DECL R1 ;BACK UP A CHAR
52 51 D1 018B 296 CMPL R1,R2 ;CHECK FOR START OF LINE
61 24 13 018E 297 BEQL 90$ ;IF FOUND, THEN DONE
61 20 91 0190 298 CMPB #^X20,(R1) ;SPACE?
61 F4 13 0193 299 BEQL 40$ ;YES, THEN BACK UP AGAIN
61 09 91 0195 300 CMPB #^X09,(R1) ;TAB?
61 EF 13 0198 301 BEQL 40$ ;YES, THEN BACK UP AGAIN
019A 302 :
019A 303 :
019A 304 : CHECK FOR CONTINUATION CHARACTER.

```

			019A	305	:				
	61	2D	91	019A	306		CMPB	#^A/-/, (R1)	:CONTINUATION?
		15	12	019D	307		BNEQ	90\$:NO, THEN DONE
	68	AB	20	A8	019F	308	BISW	#PRC_M_IND,PRC_W_FLAGS(R11)	:IGNORE @ SIGNS
		OE	6E	E9	01A3	309	BLBC	(SP),90\$:IF DATA, THEN DONE
			FE	57	30	01A4	BSBW	DCL\$INPUT	:GET CONTINUATION LINE
	68	AB	20	AA	01A9	311	BICW	#PRC_M_IND,PRC_W_FLAGS(R11)	:RESTORE @ SIGNS
02	68	AB	04	E1	01AD	312	BBC	#PRC_V_GOTO,PRC_W_FLAGS(R11),90\$:BRANCH IF EOF
			B0	11	01B2	313	BRB	2\$:FLUSH IT TOO
					01B4	314			
	5E	04	C0	01B4	315	90\$:	ADDL	#4,SP	:REMOVE TEMP. FLAG FROM STACK
				01B7	316		STATUS	NORMAL	:SET NORMAL STATUS
			05	01BE	317		RSB		:RETURN

```

01BF 319      .SBTTL  ALLOCATE A LABEL SYMBOL TABLE ENTRY
01BF 320      :+
01BF 321      : DCL$ALLOC_LABEL - ALLOCATE A LABEL SYMBOL TABLE ENTRY
01BF 322      :
01BF 323      : THIS ROUTINE IS CALLED TO ALLOCATE A LABEL SYMBOL TABLE ENTRY.
01BF 324      :
01BF 325      : INPUTS:
01BF 326      :
01BF 327      :     R1/R2 = LABEL DESCRIPTOR
01BF 328      :     R11 = ADDRESS OF PRC DATA STRUCTURE
01BF 329      :
01BF 330      : OUTPUTS:
01BF 331      :
01BF 332      :     R0 = STATUS
01BF 333      :     R4/R5 = LABEL DESCRIPTOR
01BF 334      :     R6 = SYMBOL ADDRESS
01BF 335      :
01BF 336      : CTRL/Y'S ARE ASSUMED TO HAVE BEEN DISABLED.
01BF 337      :
01BF 338      : -
01BF 339      DCL$ALLOC_LABEL::
55 7E 51 7D 01BF 340      MOVQ   R1, -(SP)
53 30 AB 9E 01C2 341      MOVAB  PRC_Q_LABEL(R11),R5
50 53 51 7D 01C6 342      MOVQ   R1,R3
52 14 AB D0 01C9 343      MOVL  PRC_L_INDINPRAB(R11),R0
51 10 A0 9E 01CD 344      MOVAB  RAB$W_RFA(R0),R2
50 51 06 9A 01D1 345      MOVZBL #6,R1
50 03 D0 01D4 346      MOVL  #SYM_K_LABEL,R0
56 FE26' 30 01D7 347      BSBW  DCL$ALLOCSYM
54 51 D0 01DA 348      MOVL  R1,R6
54 8E 7D 01DD 349      MOVQ  (SP)+,R4
05 01E0 350      RSB
                                :
                                : SAVE R1/R2
                                : SET ADDRESS OF TABLE LIST HEAD
                                : SET LABEL NAME
                                : GET RAB ADDRESS
                                : SET ADDRESS OF RFA
                                : SET LENGTH OF RFA
                                : SET SYMBOL TYPE
                                : CREATE THE LABEL ENTRY
                                : GET SYMBOL ADDRESS
                                : RESTORE R1/R2
                                :

```

```
01E1 352 .SBTTL DEALLOCATE GOTO CONTROL BLOCK
01E1 353 :+
01E1 354 : DCL$DEALGOTO - DEALLOCATE GOTO CONTROL BLOCK
01E1 355 :
01E1 356 : THIS ROUTINE IS CALLED TO DEALLOCATE AN OUTSTANDING GOTO CONTROL
01E1 357 : BLOCK WHEN IT IS SATISFIED OR AN END OF FILE IS REACHED WITHOUT
01E1 358 : FINDING THE LABEL.
01E1 359 :
01E1 360 :-
01E1 361 DCL$DEALGOTO::
50 60 AB D0 01E1 362 MOVL PRC_L_SYMBOL(R11),R0 : ADDRESS OF GOTO CONTROL BLOCK
51 60 9A 01E5 363 MOVZBL (R0),R1 : GET LENGTH AND POINT AT START OF BLOCK
FE13' D6 01E8 364 INCL R1 : INCLUDE LABEL LENGTH BYTE
60 AB D4 01EA 365 BSBW DCL$DEADYNMEM : DEALLOCATE THE BLOCK
05 01ED 366 CLRL PRC_L_SYMBOL(R11) : CLEAR ADDRESS OF GOTO CONTROL BLOCK
01F0 367 RSB :
01F1 368
01F1 369 .END
```

GOTO
Symbol table

- GOTO COMMAND EXECUTION

E 8

15-SEP-1984 23:49:46 VAX/VMS Macro V04-00
4-SEP-1984 23:40:47 [DCL.SRC]GOTO.MAR;1

```

SS.TMP1      = 00000001
SS.TMP2      = 00000067
CLIS_NORMAL  = 00030001
CLIS_SYMOVF  = 00038138
CLIS_USGOTO  = 00038148
DCLSALLDYNMEM ***** X 02
DCLSALLOCSYM ***** X 02
DCLSALLOC_LABEL 000001BF RG 02
DCLSDEADYMEM ***** X 02
DCLSDEALGOTO 000001E1 RG 02
DCLSDISABLE ***** X 02
DCLSGETDVAL ***** X 02
DCLSGOTO     00000018 RG 02
DCL$INPUT    ***** X 02
DCL$RESTART ***** X 02
DCL$SEARCHT ***** X 02
DEV$V_RND    = 0000001C
FLUSH        00000159 R 02
PRC_B_CONTINUE 000000F3
PRC_B_DEFRADIX 000000AE
PRC_B_EXMDEPMOD 000000AD
PRC_B_EXMDEPWID 000000AC
PRC_B_EXONLYL 0000012D
PRC_B_FLAGS2 000000AF
PRC_B_IMGFLAG 00000078
PRC_B_OUTFLAGS 0000012C
PRC_B_PROMPTLEN 000000F0
PRC_C_LENGTH 00000534
PRC_G_COMMANDS 00000133
PRC_G_PROMPT 000000F4
PRC_K_LENGTH 00000534
PRC_L_CURRKEY 00000048
PRC_L_EXMDEPADR 000000A8
PRC_L_EXTARG 00000094
PRC_L_EXTBLK 0000008C
PRC_L_EXTCOD 0000009C
PRC_L_EXTHND 00000090
PRC_L_EXTPRM 00000098
PRC_L_IDFLNK 0000008C
PRC_L_IMGACTSTS 00000080
PRC_L_INDCLOCK 0000007C
PRC_L_INDEPTH 0000005C
PRC_L_INDFAB 0000001C
PRC_L_INDINPRAB 00000014
PRC_L_INDOUTRAB 00000018
PRC_L_INPRAB 00000008
PRC_L_LASTKEY 0000004C
PRC_L_LSTSTATUS 00000080
PRC_L_ONCTLY 00000088
PRC_L_ONERROR 0000006C
PRC_L_OUTOFBAND 000000B4
PRC_L_OUTRAB 0000000C
PRC_L_OUTRABCTX 00000118
PRC_L_PPFLIST 00000070
PRC_L_RECALLPTR 0000012F
PRC_L_RESTART 00000058
PRC_L_SAVAP 00000000

```

```

PRC_L_SAVFP 00000004
PRC_L_SEVERITY 00000050
PRC_L_SPWN 000000C0
PRC_L_STACKLM 000000A4
PRC_L_STACKPT 000000A0
PRC_L_STATUS 00000054
PRC_L_STS 00000084
PRC_L_STV 00000088
PRC_L_SYMBOL 00000060
PRC_L_TMBX 00000074
PRC_L_TRMLIST 00000010
PRC_M_GOTO = 00000010
PRC_M_IND = 00000020
PRC_Q_ALLOCREG 00000020
PRC_Q_COMMAND 000000E0
PRC_Q_FLUSHTIME C00000D0
PRC_Q_GLOBAL 00000028
PRC_Q_IMAGENAME 000000D8
PRC_Q_KEYPAD 00000040
PRC_Q_LABEL 00000030
PRC_Q_LOCAL 00000038
PRC_Q_SAVEPRIV 000000E8
PRC_T_OUTDVI 0000011C
PRC_V_GOTO = 00000004
PRC_W_ASTIOSB 000000C6
PRC_W_ASTRETN 000000C8
PRC_W_ASTSTATUS 000000C4
PRC_W_ATTMBX 0000007A
PRC_W_FLAGS 00000068
PRC_W_INPCHAN 00000064
PRC_W_ONLEVEL 0000006A
PRC_W_OUTIFI 00000114
PRC_W_OUTISI 00000116
PRC_W_OUTMBXCHN 000000CA
PRC_W_OUTMBXREF 000000CE
PRC_W_OUTMBXSIZ 000000CC
PRC_W_PMPCTRL 000000F1
PRC_W_WAITIOSB 00000066
RAB$B_RAC = 0000001E
RAB$C_RFA = 00000002
RAB$C_SEQ = 00000000
RAB$L_CTX = 00000018
RAB$W_RFA = 00000010
ST$M_INHIB_MSG = 10000000
SYM_B_FLAGS 0000000B
SYM_B_NONUNIQUE 0000000B
SYM_B_TYPE 0000000A
SYM_K_LABEL = 00000003
SYM_L_BL = 00000004
SYM_L_FL 00000000
SYM_T_SYMBOL 0000000C
SYM_W_SIZE 00000008
SY$FIND ***** GX 02
TERM 00000000 R 02
TERMEMD 00000018 R 02
WRK_B_CMDOPT FFFFFFF3
WRK_B_MAXPARM FFFFFFFD0

```

GOTO
Symbol table

- GOTO COMMAND EXECUTION

F 8

15-SEP-1984 23:49:46 VAX/VMS Macro V04-00
4-SEP-1984 23:40:47 [DCL.SRC]GOTO.MAR;1

Page 12
(6)

```

WRK_B_MINPAM          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_G_BUFFER         FFFFF492
WRK_G_INPBUF         FFFFF896
WRK_G_RESULT         FFFFF9B6
WRK_K_LENGTH         FFFFF486
WRK_L_CHARPTR        FFFFF48E
WRK_L_DISALLOW       FFFFFFFE6
WRK_L_ERRORRTN       FFFFF9AE
WRK_L_EXPANDPTR      FFFFF486
WRK_L_IMAGE          FFFFFFFE2
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_RSLEND         FFFFFFFB6
WRK_L_RSLNXT         FFFFFFFBA
WRK_L_SAVAP          FFFFFFFF8
WRK_L_SAVFP          FFFFFFFFC
WRK_L_SAVSP          FFFFFFFF4
WRK_L_SIGNALRTN     FFFFFFFD6
WRK_L_SPECRTN        FFFFF9B2
WRK_L_TAB_VEC        FFFFFFFDE
WRK_L_VERB           FFFFFFFBE
WRK_V_NOSTAT         = 00000008
WRK_W_FLAGS          FFFFFFFF0
WRK_W_FLAGS2         FFFFFFFF2
WRK_W_IMGCHAN        FFFFFFFEE
WRK_W_PMPTLEN        FFFFF99E
_$$_                 = 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
\$AB\$\$	FFFFFFFC (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
DCL\$ZCODE	000001F1 (497.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	10	00:00:00.07	00:00:00.81
Command processing	81	00:00:00.67	00:00:05.65
Pass 1	230	00:00:07.85	00:00:23.49
Symbol table sort	0	00:00:00.83	00:00:02.48
Pass 2	65	00:00:01.48	00:00:05.15
Symbol table output	18	00:00:00.14	00:00:00.32
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	405	00:00:11.07	00:00:37.93

The working set limit was 1200 pages.
39235 bytes (77 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 661 non-local and 23 local symbols.
369 source lines were read in Pass 1, producing 14 object records in Pass 2.
37 pages of virtual memory were used to define 22 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	8
TOTALS (all libraries)	16

839 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

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

SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT
SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT	SCREENSHOT

GETKEYNAM
LIS

GOTO
LIS

EXPRESS
LIS

HANDLE
LIS

IMAGECTRL
LIS

INITIAL
LIS

INDIRECT
LIS

FILECMDS
LIS

IF
LIS

IMAGEXECT
LIS