


```
CCCCCCCC 000000 BBBB BBBB KK KK EEEEEEEEE YY YY
CCCCCCCC 000000 BBBB BBBB KK KK EEEEEEEEE YY YY
CC CC 00 00 BB BB KK KK EE YY YY
CC CC 00 00 BB BB KK KK EE YY YY
CC CC 00 00 BB BB KK KK EE YY YY
CC CC 00 00 BB BB KK KK EE YY YY
CC CC 00 00 BB BB KK KK EE YY YY
CC CC 00 00 BB BB KK KK EE YY YY
CC CC 00 00 BB BB KK KK EE YY YY
CCCCCCCC 000000 BBBB BBBB KK KK EEEEEEEEE YY YY
CCCCCCCC 000000 BBBB BBBB KK KK EEEEEEEEE YY YY
.....
```

```
LL IIIII SSSSSSS
LL IIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SSSSS
LL II SSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLL SSSSSSS
LLLLLLLLL SSSSSSS
```

(2) 46
(3) 84
(4) 298

DECLARATIONS
Table for COBOL defined KEY sequences
COB\$\$\$CONTROL_KEY - Translate terminator to COBOL sequence


```
0000 1      .TITLE COB$$$CONTROL_KEY - Key translation
0000 2      .IDENT /1-001/ ; File: COBKEY.MAR Edit: DG1001
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: Run-Time Library COBOL support
0000 31
0000 32      ABSTRACT:
0000 33
0000 34          This module contains a routine which converts
0000 35          terminator sequences to COBOL defined sequences
0000 36
0000 37      ENVIRONMENT: Runs at any access mode, AST Reentrant
0000 38
0000 39      AUTHOR: Debess Grabazs, CREATION DATE: 14-June-1983
0000 40
0000 41      MODIFIED BY:
0000 42
0000 43      1-001 - Original. DG 14-June-1983
0000 44      --
```

```
0000 46 .SBTTL DECLARATIONS
0000 47 :
0000 48 : LIBRARY MACRO CALLS:
0000 49 :
0000 50 $DSCDEF ; Get descriptor definitions
0000 51 $SMGDEF ; Get terminator codes
0000 52 :
0000 53 :
0000 54 : EXTERNAL DECLARATIONS:
0000 55 :
0000 56 .DSABL GBL ; Force all external symbols to be declared
0000 57 .EXTRN SMG$$TERM_TO_KEYCODE ; Translate terminator to key code
0000 58 :
0000 59 : MACROS:
0000 60 :
0000 61 : NONE
0000 62 :
0000 63 : EQUATED SYMBOLS:
0000 64 :
0000 65 :
00000009 0000 66 K_TAB = 9
0000000D 0000 67 K_RETURN = 13
0000001A 0000 68 K_CTRLZ = 26
0000008F 0000 69 K_SS3 = 143
0000009B 0000 70 K_CSI = 155
0000 71 :
0000 72 :
0000 73 : OWN STORAGE:
0000 74 :
0000 75 : NONE
0000 76 :
0000 77 : PSECT DECLARATIONS:
0000 78 :
0000 79 :
00000000 0000 80 .PSECT _SMG$CODE PIC, USR, CON, REL, LCL, SHR, -
0000 81 EXE, RD, NOWRT, LONG
0000 82
```


TABLE:			.SBTTL Table for COBOL defined KEY sequences	
	0000	84		
	0000	85		
	0000	86		
0112	0000	87	.WORD	SMG\$K_TRM_UP
0002	0002	88	.WORD	2
9B	0004	89	.BYTE	K_CSI
41	0005	90	.ASCII	"A"
0113	0006	91	.WORD	SMG\$K_TRM_DOWN
0002	0008	92	.WORD	2
9B	000A	93	.BYTE	K_CSI
42	000B	94	.ASCII	"B"
0114	000C	95	.WORD	SMG\$K_TRM_RIGHT
0002	000E	96	.WORD	2
9B	0010	97	.BYTE	K_CSI
43	0011	98	.ASCII	"C"
0114	0012	99	.WORD	SMG\$K_TRM_LEFT
0002	0014	100	.WORD	2
9B	0016	101	.BYTE	K_CSI
44	0017	102	.ASCII	"D"
010E	0018	103	.WORD	SMG\$K_TRM_ENTER
0002	001A	104	.WORD	2
8F	001C	105	.BYTE	K_SS3
4D	001D	106	.ASCII	"M"
0100	001E	107	.WORD	SMG\$K_TRM_PF1
0002	0020	108	.WORD	2
8F	0022	109	.BYTE	K_SS3
50	0023	110	.ASCII	"P"
0101	0024	111	.WORD	SMG\$K_TRM_PF2
0002	0026	112	.WORD	2
8F	0028	113	.BYTE	K_SS3
51	0029	114	.ASCII	"Q"
0102	002A	115	.WORD	SMG\$K_TRM_PF3
0002	002C	116	.WORD	2
8F	002E	117	.BYTE	K_SS3
52	002F	118	.ASCII	"R"
0103	0030	119	.WORD	SMG\$K_TRM_PF4
0002	0032	120	.WORD	2
8F	0034	121	.BYTE	K_SS3
53	0035	122	.ASCII	"S"
0110	0036	123	.WORD	SMG\$K_TRM_COMMA
0002	0038	124	.WORD	2
8F	003A	125	.BYTE	K_SS3
6C	003B	126	.ASCII	"T"
010F	003C	127	.WORD	SMG\$K_TRM_MINUS
0002	003E	128	.WORD	2
8F	0040	129	.BYTE	K_SS3
6D	0041	130	.ASCII	"m"
0111	0042	131	.WORD	SMG\$K_TRM_PERIOD
0002	0044	132	.WORD	2
8F	0046	133	.BYTE	K_SS3
6E	0047	134	.ASCII	"n"
0104	0048	135	.WORD	SMG\$K_TRM_KP0
0002	004A	136	.WORD	2
8F	004C	137	.BYTE	K_SS3
70	004D	138	.ASCII	"p"
0105	004E	139	.WORD	SMG\$K_TRM_KP1
0002	0050	140	.WORD	2

8F	0052	141	.BYTE	K_SS3
71	0053	142	.ASCII	..q..
0106	0054	143	.WORD	SMG\$K_TRM_KP2
0002	0056	144	.WORD	2
8F	0058	145	.BYTE	K_SS3
72	0059	146	.ASCII	..r..
0107	005A	147	.WORD	SMG\$K_TRM_KP3
0002	005C	148	.WORD	2
8F	005E	149	.BYTE	K_SS3
73	005F	150	.ASCII	..s..
0108	0060	151	.WORD	SMG\$K_TRM_KP4
0002	0062	152	.WORD	2
8F	0064	153	.BYTE	K_SS3
74	0065	154	.ASCII	..t..
0109	0066	155	.WORD	SMG\$K_TRM_KP5
0002	0068	156	.WORD	2
8F	006A	157	.BYTE	K_SS3
75	006B	158	.ASCII	..u..
010A	006C	159	.WORD	SMG\$K_TRM_KP6
0002	006E	160	.WORD	2
8F	0070	161	.BYTE	K_SS3
76	0071	162	.ASCII	..v..
010B	0072	163	.WORD	SMG\$K_TRM_KP7
0002	0074	164	.WORD	2
8F	0076	165	.BYTE	K_SS3
77	0077	166	.ASCII	..w..
010C	0078	167	.WORD	SMG\$K_TRM_KP8
0002	007A	168	.WORD	2
8F	007C	169	.BYTE	K_SS3
78	007D	170	.ASCII	..x..
010D	007E	171	.WORD	SMG\$K_TRM_KP9
0002	0080	172	.WORD	2
8F	0082	173	.BYTE	K_SS3
79	0083	174	.ASCII	..y..
0137	0084	175	.WORD	SMG\$K_TRM_FIND
0003	0086	176	.WORD	3
9B	0088	177	.BYTE	K_CSI
31	0089	178	.ASCII	..T..
7E	008A	179	.ASCII	..-..
0138	008B	180	.WORD	SMG\$K_TRM_INSERT_HERE
0003	008D	181	.WORD	3
9B	008F	182	.BYTE	K_CSI
32	0090	183	.ASCII	..2..
7E	0091	184	.ASCII	..-..
0139	0092	185	.WORD	SMG\$K_TRM_REMOVE
0003	0094	186	.WORD	3
9B	0096	187	.BYTE	K_CSI
33	0097	188	.ASCII	..3..
7E	0098	189	.ASCII	..-..
013A	0099	190	.WORD	SMG\$K_TRM_SELECT
0003	009B	191	.WORD	3
9B	009D	192	.BYTE	K_CSI
34	009E	193	.ASCII	..4..
7E	009F	194	.ASCII	..-..
013B	00A0	195	.WORD	SMG\$K_TRM_PREV_SCREEN
0003	00A2	196	.WORD	3
9B	00A4	197	.BYTE	K_CSI

35	00A5	198	.ASCII	..5..
7E	00A6	199	.ASCII	..2..
013C	00A7	200	.WORD	SMG\$K_TRM_NEXT_SCREEN
0003	00A9	201	.WORD	3
9B	00AB	202	.BYTE	K_CSI
36	00AC	203	.ASCII	..8..
7E	00AD	204	.ASCII	..3..
011E	00AE	205	.WORD	SMG\$K_TRM_F6
0004	00B0	206	.WORD	4
9B	00B2	207	.BYTE	K_CSI
31	00B3	208	.ASCII	..T..
37	00B4	209	.ASCII	..7..
7E	00B5	210	.ASCII	..-..
011F	00B6	211	.WORD	SMG\$K_TRM_F7
0004	00B8	212	.WORD	4
9B	00BA	213	.BYTE	K_CSI
31	00BB	214	.ASCII	..T..
38	00BC	215	.ASCII	..8..
7E	00BD	216	.ASCII	..-..
0120	00BE	217	.WORD	SMG\$K_TRM_F8
0004	00C0	218	.WORD	4
9B	00C2	219	.BYTE	K_CSI
31	00C3	220	.ASCII	..T..
39	00C4	221	.ASCII	..9..
7E	00C5	222	.ASCII	..-..
0121	00C6	223	.WORD	SMG\$K_TRM_F9
0004	00C8	224	.WORD	4
9B	00CA	225	.BYTE	K_CSI
32	00CB	226	.ASCII	..2..
30	00CC	227	.ASCII	..0..
7E	00CD	228	.ASCII	..-..
0122	00CE	229	.WORD	SMG\$K_TRM_F10
0004	00D0	230	.WORD	4
9B	00D2	231	.BYTE	K_CSI
32	00D3	232	.ASCII	..2..
31	00D4	233	.ASCII	..1..
7E	00D5	234	.ASCII	..-..
0123	00D6	235	.WORD	SMG\$K_TRM_F11
0004	00D8	236	.WORD	4
9B	00DA	237	.BYTE	K_CSI
32	00DB	238	.ASCII	..2..
33	00DC	239	.ASCII	..3..
7E	00DD	240	.ASCII	..-..
0124	00DE	241	.WORD	SMG\$K_TRM_F12
0004	00E0	242	.WORD	4
9B	00E2	243	.BYTE	K_CSI
32	00E3	244	.ASCII	..2..
34	00E4	245	.ASCII	..4..
7E	00E5	246	.ASCII	..-..
0125	00E6	247	.WORD	SMG\$K_TRM_F13
0004	00E8	248	.WORD	4
9B	00EA	249	.BYTE	K_CSI
32	00EB	250	.ASCII	..2..
35	00EC	251	.ASCII	..5..
7E	00ED	252	.ASCII	..-..
0126	00EE	253	.WORD	SMG\$K_TRM_F14
0004	00F0	254	.WORD	4

9B	00F2	255	.BYTE	K_CSI
32	00F3	256	.ASCII	..2..
36	00F4	257	.ASCII	..6..
7E	00F5	258	.ASCII	..-..
0127	00F6	259	.WORD	SMG\$K_TRM_HELP
0004	00F8	260	.WORD	4
9B	00FA	261	.BYTE	K_CSI
32	00FB	262	.ASCII	..2..
38	00FC	263	.ASCII	..8..
7E	00FD	264	.ASCII	..-..
0128	00FE	265	.WORD	SMG\$K_TRM_DO
0004	0100	266	.WORD	4
9B	0102	267	.BYTE	K_CSI
32	0103	268	.ASCII	..2..
39	0104	269	.ASCII	..9..
7E	0105	270	.ASCII	..-..
0129	0106	271	.WORD	SMG\$K_TRM_F17
0004	0108	272	.WORD	4
9B	010A	273	.BYTE	K_CSI
33	010B	274	.ASCII	..3..
31	010C	275	.ASCII	..1..
7E	010D	276	.ASCII	..-..
012A	010E	277	.WORD	SMG\$K_TRM_F18
0004	0110	278	.WORD	4
9B	0112	279	.BYTE	K_CSI
33	0113	280	.ASCII	..3..
32	0114	281	.ASCII	..2..
7E	0115	282	.ASCII	..-..
012B	0116	283	.WORD	SMG\$K_TRM_F19
0004	0118	284	.WORD	4
9B	011A	285	.BYTE	K_CSI
33	011B	286	.ASCII	..3..
33	011C	287	.ASCII	..3..
7E	011D	288	.ASCII	..-..
012C	011E	289	.WORD	SMG\$K_TRM_F20
0004	0120	290	.WORD	4
9B	0122	291	.BYTE	K_CSI
33	0123	292	.ASCII	..3..
34	0124	293	.ASCII	..4..
7E	0125	294	.ASCII	..-..
0000	0126	295	.WORD	0
	0128	296		

; End of table

```

0128 298 .SBTTL COB$$$CONTROL_KEY - Translate terminator to COBOL sequence
0128 299 :++
0128 300 : FUNCTIONAL DESCRIPTION:
0128 301 :
0128 302 : COB$$$CONTROL_KEY translates a terminator character sequence
0128 303 : to a COBOL defined sequence.
0128 304 :
0128 305 : CALLING SEQUENCE:
0128 306 :
0128 307 : CALL COB$$$CONTROL_KEY (terminator.rt.r, term_length.rl.v, key.wt.ds)
0128 308 :
0128 309 :
0128 310 : FORMAL PARAMETERS:
0128 311 :
00000004 0128 312 : terminator = 4 ; The terminator string, passed by reference.
0128 313 :
00000008 0128 314 : term_length = 8 ; The length of the terminator string, passed
0128 315 : by immediate value.
0128 316 :
0000000C 0128 317 : key = 12 ; Destination of receiving field of the
0128 318 : control key, passed by descriptor.
0128 319 :
0128 320 :
0128 321 : IMPLICIT INPUTS:
0128 322 :
0128 323 : NONE
0128 324 :
0128 325 : IMPLICIT OUTPUTS:
0128 326 :
0128 327 : NONE
0128 328 :
0128 329 : COMPLETION STATUS:
0128 330 :
0128 331 : NONE
0128 332 :
0128 333 : SIDE EFFECTS:
0128 334 :
0128 335 : KEY gets filled in with the appropriate COBOL defined sequence.
0128 336 :
0128 337 :--
0001 0128 338 :
0001 20 0128 339 ONE: .WORD 1 ..
0128 340 SPACE: .ASCII .. ..
0004 0128 341
0004 0128 342 .ENTRY COB$$$CONTROL_KEY, ^M<R2>
0128 343
0128 344 :+
0128 345 : Space fill KEY to initialize
0128 346 :-
0128 347 :
0128 348 MOVL key(AP), R0
0128 349 MOVC5 ONE, SPACE, #^A/ /, DSC$W_LENGTH(R0), @DSC$A_POINTER(R0)
0138 350
013A 351 :+
013A 352 : Translate terminator sequence to key code
013A 353 :-

```

60 20 F4 50 0C AC D0
F4 AF 2C
04 B0


```
00000000'EF 08 AC DD 013A 354
04 BC DD 013A 355          PUSHL term_length(AP)
02 FB DD 013D 356          PUSHL @terminator(AP)
0140 FB 0140 357          CALLS #2, SMG$$TERM_TO_KEYCODE
0147 358
0147 359 ;+
0147 360 ; Search the table to determine what COBOL defined sequence to put in KEY.
0147 361 ; R1 will be loaded with the table address and R0 is the key code returned
0147 362 ; by SMG$$TERM_TO_KEYCODE.
0147 363 ; -
0147 364
51 FEB5 CF 9E 0147 365          MOVAB W^TABLE, R1
014C 366
52 81 3C 014C 367 10$: MOVZWL (R1)+, R2          ; Get word to compare against
1C 13 014F 368          BEQL NOTRANS          ; End of table?
50 52 B1 0151 369          CMPW R2, R0          ; Compare character
09 13 0154 370          BEQL FOUND          ; End if found
52 61 02 A1 0156 371          ADDW3 #2, (R1), R2      ; Skip over contents
51 52 A0 015A 372          ADDW2 R2, R1
ED 11 015D 373          BRB 10$          ; Repeat until found or table end
015F 374
015F 375 ;+
015F 376 ; Translation found. Store the counted sequence which is in the bytes
015F 377 ; addressed by R1
015F 378 ; -
015F 379
015F 380 FOUND:
04 B0 50 02 A1 0C AC D0 015F 381          MOVL KEY(AP), R0
01 61 28 0163 382          MOVC3 (R1), 2(R1), @DSC$A_POINTER(R0) ; Put sequence in KEY
50 01 D0 0169 383          MOVL #1, R0          ; Signal success
04 01 04 016C 384          RET
016D 385
016D 386 ;+
016D 387 ; We get here if the terminator has no translation
016D 388 ; -
016D 389
50 01FF 8F 3C 016D 390 NOTRANS:
50 00 D0 016D 391          MOVZWL #SMG$K_TRM_UNKNOWN, R0 ; Unknown terminator
04 0172 392          MOVL #0, R0          ; Signal failure - unknown key
0175 393          RET          ; Return terminator code
0176 394
0176 395          .END          ; End of module COB$$$CONTROL_KEY
```

COB\$\$\$CONTROL_KEY
Symbol table

- Key translation

H 13

15-SEP-1984 23:45:57 VAX/VMS Macro V04-00
6-SEP-1984 10:47:34 [COBRTL.SRC]COBKEY.MAR;1

Page 9
(4)

TERM_LENGTH

= 00000008

COB\$\$\$CONTROL_KEY	= 0000012B	RG	02
DSC\$A_POINTER	= 00000004		
DSC\$W_LENGTH	= 00000000		
FOUND	= 0000015F	R	02
KEY	= 0000000C		
K_CSI	= 0000009B		
K_SS3	= 0000008F		
NOTRANS	= 0000016D	R	02
ONE	= 00000128	R	02
SMG\$\$\$TERM_TO_KEYCODE	*****	X	00
SMG\$K_TRM_COMMA	= 00000110		
SMG\$K_TRM_DO	= 00000128		
SMG\$K_TRM_DOWN	= 00000113		
SMG\$K_TRM_ENTER	= 0000010E		
SMG\$K_TRM_F10	= 00000122		
SMG\$K_TRM_F11	= 00000123		
SMG\$K_TRM_F12	= 00000124		
SMG\$K_TRM_F13	= 00000125		
SMG\$K_TRM_F14	= 00000126		
SMG\$K_TRM_F17	= 00000129		
SMG\$K_TRM_F18	= 0000012A		
SMG\$K_TRM_F19	= 0000012B		
SMG\$K_TRM_F20	= 0000012C		
SMG\$K_TRM_F6	= 0000011E		
SMG\$K_TRM_F7	= 0000011F		
SMG\$K_TRM_F8	= 00000120		
SMG\$K_TRM_F9	= 00000121		
SMG\$K_TRM_FIND	= 00000137		
SMG\$K_TRM_HELP	= 00000127		
SMG\$K_TRM_INSERT_HERE	= 00000138		
SMG\$K_TRM_KP0	= 00000104		
SMG\$K_TRM_KP1	= 00000105		
SMG\$K_TRM_KP2	= 00000106		
SMG\$K_TRM_KP3	= 00000107		
SMG\$K_TRM_KP4	= 00000108		
SMG\$K_TRM_KP5	= 00000109		
SMG\$K_TRM_KP6	= 0000010A		
SMG\$K_TRM_KP7	= 0000010B		
SMG\$K_TRM_KP8	= 0000010C		
SMG\$K_TRM_KP9	= 0000010D		
SMG\$K_TRM_LEFT	= 00000114		
SMG\$K_TRM_MINUS	= 0000010F		
SMG\$K_TRM_NEXT_SCREEN	= 0000013C		
SMG\$K_TRM_PERIOD	= 00000111		
SMG\$K_TRM_PF1	= 00000100		
SMG\$K_TRM_PF2	= 00000101		
SMG\$K_TRM_PF3	= 00000102		
SMG\$K_TRM_PF4	= 00000103		
SMG\$K_TRM_PREV_SCREEN	= 0000013B		
SMG\$K_TRM_REMOVE	= 00000139		
SMG\$K_TRM_RIGHT	= 00000115		
SMG\$K_TRM_SELECT	= 0000013A		
SMG\$K_TRM_UNKNOWN	= 000001FF		
SMG\$K_TRM_UP	= 00000112		
SPACE	= 0000012A	R	02
TABLE	= 00000000	R	02
TERMINATOR	= 00000004		

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

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR
_SMG\$CODE	00000176 (374.)	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	29	00:00:00.05	00:00:01.36
Command processing	106	00:00:00.35	00:00:03.95
Pass 1	184	00:00:02.72	00:00:12.21
Symbol table sort	0	00:00:00.33	00:00:00.80
Pass 2	74	00:00:00.69	00:00:04.05
Symbol table output	8	00:00:00.05	00:00:00.15
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	406	00:00:04.21	00:00:22.54

The working set limit was 1050 pages.
20001 bytes (40 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 349 non-local and 1 local symbols.
395 source lines were read in Pass 1, producing 13 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5

397 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:COBKEY/OBJ=OBJ\$:COBKEY MSRC\$:COBKEY/UPDATE=(ENH\$:COBKEY)

0063 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

