

|        |        |                |                  |                  |              |              |     |
|--------|--------|----------------|------------------|------------------|--------------|--------------|-----|
| MMM    | MMM    | TTTTTTTTTTTTTT | AAAAAAAAA        | AAAAAAAAA        | CCCCCCCCCCCC | PPPPPPPPPPPP |     |
| MMM    | MMM    | TTTTTTTTTTTTTT | AAAAAAAAA        | AAAAAAAAA        | CCCCCCCCCCCC | PPPPPPPPPPPP |     |
| MMM    | MMM    | TTTTTTTTTTTTTT | AAAAAAAAA        | AAAAAAAAA        | CCCCCCCCCCCC | PPPPPPPPPPPP |     |
| MMMMMM | MMMMMM | TTT            | AAA              | AAA              | CCC          | PPP          | PPP |
| MMMMMM | MMMMMM | TTT            | AAA              | AAA              | CCC          | PPP          | PPP |
| MMMMMM | MMMMMM | TTT            | AAA              | AAA              | CCC          | PPP          | PPP |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          | PPP |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          | PPP |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          | PPP |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          | PPP |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPPPPPPPPPPP |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPPPPPPPPPPP |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPPPPPPPPPPP |     |
| MMM    | MMM    | TTT            | AAAAAAAAAAAAAAAA | AAAAAAAAAAAAAAAA | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAAAAAAAAAAAAAAA | AAAAAAAAAAAAAAAA | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAAAAAAAAAAAAAAA | AAAAAAAAAAAAAAAA | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCC          | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCCCCCCCCCCC | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCCCCCCCCCCC | PPP          |     |
| MMM    | MMM    | TTT            | AAA              | AAA              | CCCCCCCCCCCC | PPP          |     |

|          |    |           |           |            |          |      |    |      |
|----------|----|-----------|-----------|------------|----------|------|----|------|
| DDDDDDDD |    | AAAAAA    | TTTTTTTTT | EEEEEEEEEE | CCCCCCCC | NN   | NN |      |
| DDDDDDDD |    | AAAAAA    | TTTTTTTTT | EEEEEEEEEE | CCCCCCCC | NN   | NN |      |
| DD       | DD | AA        | TT        | EE         | CC       | NN   | NN |      |
| DD       | DD | AA        | TT        | EE         | CC       | NN   | NN |      |
| DD       | DD | AA        | TT        | EE         | CC       | NNNN | NN |      |
| DD       | DD | AA        | TT        | EE         | CC       | NNNN | NN |      |
| DD       | DD | AA        | TT        | EEEEEEEE   | CC       | NN   | NN |      |
| DD       | DD | AA        | TT        | EEEEEEEE   | CC       | NN   | NN |      |
| DD       | DD | AAAAAAAAA | TT        | EE         | CC       | NN   | NN |      |
| DD       | DD | AAAAAAAAA | TT        | EE         | CC       | NN   | NN |      |
| DD       | DD | AA        | TT        | EE         | CC       | NN   | NN |      |
| DD       | DD | AA        | TT        | EE         | CC       | NN   | NN |      |
| DD       | DD | AA        | TT        | EE         | CC       | NN   | NN |      |
| DDDDDDDD |    | AA        | TT        | EEEEEEEEEE | CCCCCCCC | NN   | NN | .... |
| DDDDDDDD |    | AA        | TT        | EEEEEEEEEE | CCCCCCCC | NN   | NN | .... |

|            |        |          |
|------------|--------|----------|
| 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 |

```
0001 0
0002 0 MODULE DATECN ( LANGUAGE ( BLISS32 ) ,
0003          IDENT = 'V04-000'
0004          ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 *  ALL RIGHTS RESERVED.
0012 1 *
0013 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 *  TRANSFERRED.
0019 1 *
0020 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 *  CORPORATION.
0023 1 *
0024 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *****
0028 1
0029 1 ++
0030 1
0031 1 FACILITY: MTAACP
0032 1
0033 1 ABSTRACT:
0034 1
0035 1     CONVDATE_R2J converts dates from DD-MMM-YY format to (blank)YYDDD
0036 1     CONVDATE_J2R converts dates from (blank)YYDDD to DD-MMM-YY
0037 1
0038 1
0039 1 ENVIRONMENT:
0040 1
0041 1     STARLET operating system, including privileged system services
0042 1     and internal exec routines.
0043 1
0044 1 --
0045 1
0046 1
0047 1
0048 1 AUTHOR: D. H. GILLESPIE,      CREATION DATE: 31-MAY-77   9:00
0049 1
0050 1 MODIFIED BY:
0051 1
0052 1     V02-005 DMW00055      David Michael Walp      30-Nov-1981
0053 1     Fixed problem of 366 days in a none leap year when converting
0054 1     from Julian to VMS
0055 1
0056 1     V02-004 DMW00039      David Michael Walp      2-Oct-1981
0057 1     Handle Julian Date "00000" and Regular Dates less than 1900
```

```

58 0058 1 |
59 0059 1 |
60 0060 1 |
61 0061 1 |
62 0062 1 |
63 0063 1 |
64 0064 1 |
65 0065 1 |
66 0066 1 |
67 0067 1 |
68 0068 1 |
69 0069 1 |
70 0453 1 |
71 0454 1 |
72 0455 1 |
73 0456 1 |
74 0457 1 |
75 0458 1 |
76 0459 1 |
77 0460 1 |
78 0461 1 |
79 0462 1 |
80 0463 1 |
81 0464 1 |
82 0465 1 |
83 0466 1 |
84 0467 1 |
85 0468 1 |
86 0469 1 |

V02-003 REFORMAT      Maria del C. Nasr      30-Jun-1980

A0002 MCN0012      Maria del C. Nasr      15-Feb-1980  3:20
      Check for date out of range when converting to either Julian
      format or regular date

**

LIBRARY 'SYSS$LIBRARY:LIB.L32';

REQUIRE 'SRC$:MTADEF.B32';

FORWARD ROUTINE
      CONVDATE_R2J,      ! regular to julian
      CONVDATE_J2R;      ! julian to regular

EXTERNAL ROUTINE
      LIB$CVT_DTB : ADDRESSING_MODE ( ABSOLUTE ), ! convert decimal to binary
      SYSS$FAO   : ADDRESSING_MODE ( ABSOLUTE ); ! formatted ascii output

GLOBAL BIND
      ZERO_JDATE = UPLIT ( ' 00000' );

BIND
      DAYTBL = UPLIT WORD ( 0,31,59,90,120,151,181,212,243,273,304,334,365 )
      : VECTOR [,WORD],
      MONTBL = UPLIT BYTE ( 'JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC ' )
      : VECTOR [CH$ALLOCATION(48)];
```

```

88 0470 1 GLOBAL ROUTINE CONVDATE_R2J ( REGDATE, ANSIJDATE ) =
89 0471 1
90 0472 1 ++
91 0473 1
92 0474 1 FUNCTIONAL DESCRIPTION:
93 0475 1     convert from DD-MMM-YYYY to (blank)YYDDD
94 0476 1
95 0477 1
96 0478 1 CALLING SEQUENCE:
97 0479 1     CONVDATE_R2J ( ARG1, ARG2 )
98 0480 1
99 0481 1 INPUT PARAMETERS:
100 0482 1     ARG1 address of input string DD-MMM-YYYY
101 0483 1     ARG2 address of output string (blank)YYDDD
102 0484 1
103 0485 1 IMPLICIT INPUTS:
104 0486 1     none
105 0487 1
106 0488 1 OUTPUT PARAMETERS:
107 0489 1     output string pointed to by ARG2
108 0490 1
109 0491 1 IMPLICIT OUTPUTS:
110 0492 1     none
111 0493 1
112 0494 1 ROUTINE VALUE:
113 0495 1     1 if successful
114 0496 1     0 systax error
115 0497 1
116 0498 1 SIDE EFFECTS:
117 0499 1     none
118 0500 1
119 0501 1 --
120 0502 1
121 0503 2 BEGIN
122 0504 2
123 0505 2 LOCAL
124 0506 2     CONVOUTPUT      : VECTOR [2],           ! output descriptor
125 0507 2     MONTH,        ! index into month table
126 0508 2     DAYS,         ! number of days
127 0509 2     REGADR,       ! char ptr to reg date string
128 0510 2     YEAR,         ! number of year
129 0511 2     LPYEAR;       ! leap year
130 0512 2
131 0513 2 BIND
132 0514 2     BASE_YEAR  = UPLIT BYTE ( '19' ),
133 0515 2     ZERO_YEAR = UPLIT BYTE ( '0000' ),
134 0516 2     CONVCONTROL = DESCRIPTOR ( ' !AD!3ZL' ),
135 0517 2     BYEAR      = YEAR                      : BLOCK [1];
136 0518 2
137 0519 2 ! setup addr pointer and convert DD to binary # of days
138 0520 2
139 0521 2 IF (.REGDATE)<0,8> EQL ' '
140 0522 2 THEN
141 0523 2     BEGIN
142 0524 2     REGADR = .REGDATE + 1;
143 0525 2     IF NOT LIB$CVT_DTB ( 1, .REGADR, DAYS ) THEN RETURN 0;
144 0526 2     END
```

```
145 0527 2 ELSE
146 0528 IF NOT LIB$CVT_DTB ( 2, .REGDATE, DAYS ) THEN RETURN 0;
147 0529
148 0530 ! point into date to MMM
149 0531
150 0532 REGADR = .REGDATE + 3;
151 0533
152 0534 ! lookup MMM in table to discover # of days in year preceding current month
153 0535
154 0536 MONTH = 0;
155 0537 WHILE 1
156 0538 DO
157 0539 BEGIN
158 0540 IF CH$EQL ( 3, .REGADR, 3, MONTBL[.MONTH], 0 ) THEN EXITLOOP;
159 0541 MONTH = .MONTH + 1;
160 0542 IF .MONTH GEQU 12 THEN RETURN 0;
161 0543 END;
162 0544
163 0545 ! convert year
164 0546
165 0547 REGADR = .REGDATE + 7;
166 0548 IF NOT LIB$CVT_DTB ( 4, .REGADR, YEAR ) THEN RETURN 0;
167 0549
168 0550 ! determine if year is leap year
169 0551
170 0552 LPYEAR = 0;
171 0553 IF .BYEAR [ 0, 0, 2, 0 ] EQLU 0 THEN LPYEAR = 1;
172 0554
173 0555 ! decide if legal day in month
174 0556
175 0557 IF ( .DAYS GTRU ( .DAYTBL[.MONTH + 1] - .DAYTBL[.MONTH] ) )
176 0558 AND ( NOT ( .LPYEAR AND ( .MONTH EQLU 1 ) AND ( .DAYS EQLU 29 ) ) )
177 0559 THEN RETURN 0;
178 0560
179 0561 ! calculate the days into the year
180 0562
181 0563 DAYS = .DAYS + .DAYTBL[.MONTH];
182 0564 IF .MONTH GTRU 1 THEN DAYS = .DAYS + .LPYEAR;
183 0565
184 0566 ! if the year is less than 1900, then julian date must be zero
185 0567
186 0568 IF CH$LESS ( 2, .REGADR, 2, BASE_YEAR )
187 0569 THEN
188 0570 BEGIN
189 0571 CH$MOVE ( 4, ZERO_YEAR, .REGADR );
190 0572 DAYS = 0;
191 0573 END;
192 0574
193 0575 ! setup output string addr and length
194 0576
195 0577 CONVOUTPUT[0] = 6;
196 0578 CONVOUTPUT[1] = .ANSIJDATE;
197 0579
198 0580 ! setup yy char ptr
199 0581
200 0582 REGADR = .REGDATE + 9;
201 0583 2
```

```
: 202      0584 2      ! format output string
: 203      0585 2      !
: 204      0586 2      SYSS$FAO ( CONVCONTROL, 0, CONVOUTPUT, 2, .REGADR, .DAYS );
: 205      0587 2      RETURN 1;
: 206      0588 2
: 207      0589 1      END;
```

! end routine

```
.TITLE DATECN
.IDENT \V04-000\

.PSECT $CODE$,NOWRT,2

0111 00F3 00D4 00B5 0097 0078 005A 003B 001F 0000 0000 P.AAA: .ASCII \ 00000\<0><0>
52 50 41 20 52 41 4D 20 42 45 46 20 4E 41 4A 0000 P.AAB: .WORD 0, 31, 59, 90, 120, 151, 181, 212, 243, -
55 41 20 4C 55 4A 20 4E 55 4A 20 59 41 4D 20 0001C 273, 304, 334, 365
20 54 43 4F 20 50 45 53 20 47 00022 P.AAC: .ASCII \JAN FEB MAR APR MAY JUN JUL AUG SEP OCT \
20 43 45 44 20 56 4F 4E 00040
39 31 00052 P.AAD: .ASCII \NOV DEC \
00054 P.AAE: .ASCII \19\
00058 P.AAG: .ASCII \0000\
00060 P.AAF: .ASCII \ !AD!3ZL\
00064 .LONG 8
.ADDRESS P.AAG
```

```
ZERO JDATE== P.AAA
DAYTBL= P.AAB
MONTBL= P.AAC
BASE_YEAR= P.AAD
ZERO_YEAR= P.AAE
CONVCONTROL= P.AAF
.EXTRN LIB$CVT_DTB, SYSS$FAO
```

```
01FC 00000
58 00000000G 9F 9E 00002 .ENTRY CONVDATE R2J, Save R2,R3,R4,R5,R6,R7,R8 : 0470
57 94 AF 9E 00009 MOVAB @#LIB$CVT_DTB, R8
5E 10 C2 0000D MOVAB DAYTBL, R7
56 04 AC D0 00010 SUBL2 #16, SP
20 66 91 00014 MOVL REGDATE, R6 : 0521
OC 12 00017 CMPB (R6), #32
54 01 A6 9E 00019 BNEQ 1$, : 0524
4010 8F BB 0001D MOVAB 1(R6), REGADR : 0525
01 DD 00021 PUSHB #^M<R4,SP>
06 11 00023 PUSHL #1
4040 8F BB 00025 1$: PUSHB #^M<R6,SP> : 0528
02 DD 00029 PUSHL #2
68 03 FB 0002B 2$: CALLS #3, LIB$CVT_DTB
27 50 E9 0002E BLBC R0, 5$
54 03 A6 9E 00031 MOVAB 3(R6), REGADR : 0532
55 D4 00035 CLRL MONTH : 0536
1A A745 DF 00037 3$: PUSHAL MONTBL[MONTH] : 0540
64 03 29 0003B CMPC3 #3, (REGADR), @ (SP)+
09 13 0003F BEQL 4$
55 D6 00041 INCL MONTH
OC 55 D1 00043 CMPL MONTH, #12 : 0541
EF 1F 00046 BLSSU 3$ : 0542
```

|           |    |      |       |       |             |                         |                         |
|-----------|----|------|-------|-------|-------------|-------------------------|-------------------------|
| 54        | 07 | 7C   | 11    | 00048 | BRB         | 10\$                    |                         |
|           | 04 | A6   | 9E    | 0004A | 4\$: MOVAB  | 7(R6), REGADR           | 0547                    |
|           |    | AE   | 9F    | 0004E | PUSHAB      | YEAR                    | 0548                    |
|           |    | 54   | DD    | 00051 | PUSHL       | REGADR                  |                         |
|           |    | 04   | DD    | 00053 | PUSHL       | #4                      |                         |
| 68        |    | 03   | FB    | 00055 | CALLS       | #3, LIB\$CVT_DTB        |                         |
| 6B        |    | 50   | E9    | 00058 | 5\$: BLBC   | R0, 10\$                |                         |
|           |    | 51   | D4    | 0005B | CLRL        | LPYEAR                  | 0552                    |
| 03        | 04 | AE   | 93    | 0005D | BITB        | BYEAR, #3               | 0553                    |
|           |    | 03   | 12    | 00061 | BNEQ        | 6\$                     |                         |
| 51        |    | 01   | D0    | 00063 | MOVL        | #1, LPYEAR              |                         |
| 50        | 02 | A745 | 3C    | 00066 | 6\$: MOVZWL | DAYTBL+2[MONTH], R0     | 0557                    |
| 52        |    | 6745 | 3C    | 0006B | MOVZWL      | DAYTBL[MONTH], R2       |                         |
| 50        |    | 52   | C2    | 0006F | SUBL2       | R2, R0                  |                         |
| 50        |    | 6E   | D1    | 00072 | CMPL        | DAYS, R0                |                         |
|           |    | 0D   | 1B    | 00075 | BLEQU       | 7\$                     |                         |
| 4C        |    | 51   | E9    | 00077 | BLBC        | LPYEAR, 10\$            | 0558                    |
| 01        |    | 55   | D1    | 0007A | CMPL        | MONTH, #1               |                         |
|           |    | 47   | 12    | 0007D | BNEQ        | 10\$                    |                         |
| 1D        |    | 6E   | D1    | 0007F | CMPL        | DAYS, #29               |                         |
|           |    | 42   | 12    | 00082 | BNEQ        | 10\$                    |                         |
| 50        |    | 6745 | 3C    | 00084 | 7\$: MOVZWL | DAYTBL[MONTH], R0       | 0563                    |
| 6E        |    | 50   | C0    | 00088 | ADDL2       | R0, DAYS                |                         |
| 01        |    | 55   | D1    | 0008B | CMPL        | MONTH, #1               | 0564                    |
|           |    | 03   | 1B    | 0008E | BLEQU       | 8\$                     |                         |
| 6E        |    | 51   | C0    | 00090 | ADDL2       | LPYEAR, DAYS            |                         |
| 4A        | A7 | 64   | 02    | 29    | 8\$: CMPC3  | #2, (REGADR), BASE_YEAR | 0568                    |
|           |    | 06   | 1E    | 00098 | BGEQU       | 9\$                     |                         |
| 64        |    | 4C   | A7    | D0    | 0009A       | MOVL                    | ZERO_YEAR, (REGADR)     |
|           |    | 6E   | D4    | 0009E | CLRL        | DAYS                    | 0571                    |
| 08        |    | 06   | D0    | 000A0 | 9\$: MOVL   | #6, CONVOUTPUT          | 0572                    |
| OC        |    | 08   | AC    | D0    | 000A4       | MOVL                    | ANSIJDATE, CONVOUTPUT+4 |
|           |    | 09   | A6    | 9E    | 000A9       | MOVAB                   | 9(R6), REGADR           |
|           |    | 6E   | DD    | 000AD | PUSHL       | DAYS                    | 0577                    |
|           |    | 54   | DD    | 000AF | PUSHL       | REGADR                  | 0578                    |
|           |    | 02   | DD    | 000B1 | PUSHL       | #2                      | 0582                    |
|           |    | 14   | AE    | 9F    | 000B3       | PUSHAB                  | CONVOUTPUT              |
|           |    | 7E   | D4    | 000B6 | CLRL        | -(SP)                   | 0586                    |
|           |    | 58   | A7    | 9F    | 000B8       | PUSHAB                  | CONVCONTROL             |
|           |    | 06   | FB    | 000BB | CALLS       | #6, @#SYSS\$FA0         |                         |
| 00000000G | 9F | 01   | D0    | 000C2 | MOVL        | #1, R0                  | 0587                    |
|           | 50 | 04   | 000C5 | RET   |             |                         |                         |
|           |    | 50   | D4    | 000C6 | 10\$: CLRL  | R0                      | 0589                    |
|           |    | 04   | 000C8 | RET   |             |                         |                         |

; Routine Size: 201 bytes, Routine Base: \$CODE\$ + 0068

```
209 0590 1 GLOBAL ROUTINE CONVDATE_J2R ( DATEREG, DATEANSIJ ) =
210 0591 1
211 0592 1 ++
212 0593 1
213 0594 1 FUNCTIONAL DESCRIPTION:
214 0595 1     convert from (blank)YYDDD to DD-MMM-YYYY
215 0596 1
216 0597 1 CALLING SEQUENCE:
217 0598 1     CONVDATE_J2R(ARG1,ARG2)
218 0599 1
219 0600 1 INPUT PARAMETERS:
220 0601 1     ARG1 addr of output string DD-MMM-YYYY( 11 char )
221 0602 1     ARG2 addr of input string (blank)YYDDD
222 0603 1
223 0604 1 IMPLICIT INPUTS:
224 0605 1     none
225 0606 1
226 0607 1 OUTPUT PARAMETERS:
227 0608 1     output string pointed to by ARG1
228 0609 1
229 0610 1 IMPLICIT OUTPUTS:
230 0611 1     none
231 0612 1
232 0613 1 ROUTINE VALUE:
233 0614 1     2 if date of " 00000"
234 0615 1     1 if normal date
235 0616 1     0 if an invalid date
236 0617 1
237 0618 1 SIDE EFFECTS:
238 0619 1     none
239 0620 1
240 0621 1 --
241 0622 1
242 0623 2 BEGIN
243 0624 2
244 0625 2 LOCAL
245 0626 2     CONVOUTPUT      : VECTOR [2],
246 0627 2     MONTH,          : index into month table /3
247 0628 2     DAYS,          : # of days in month
248 0629 2     YEAR,         : # of years
249 0630 2     JULADR,        : addr of JULIAN char
250 0631 2     LPYEAR,       : 1 if leap year
251 0632 2     FEB29;       : 1 if FEB 29th
252 0633 2
253 0634 2 BIND
254 0635 2     CONVCONTROL = DESCRIPTOR ( '!2ZL-!AD-!4ZL' ),
255 0636 2     BYEAR      = YEAR                      : BLOCK [1];
256 0637 2
257 0638 2 ! if ANSI Julian date is zero, return with 2
258 0639 2 !
259 0640 2 IF CH$EQL ( 6, .DATEANSIJ, 6, ZERO_JDATE ) THEN RETURN 2;
260 0641 2
261 0642 2 ! setup pointer to DDD in Julian input string
262 0643 2 !
263 0644 2 JULADR = CH$PLUS ( .DATEANSIJ, 3 );
264 0645 2
265 0646 2 ! convert to binary date
```

```
266 0647 2 !
267 0648 2 !
268 0649 2 !
269 0650 2 ! backup pointer to yy and convert to binary and add century
270 0651 2 !
271 0652 2 JULADR = CH$PLUS ( .JULADR, -2 );
272 0653 2 IF NOT LIB$CVT_DTB ( 2, .JULADR, YEAR ) THEN RETURN 0;
273 0654 2 YEAR = .YEAR + 1900;
274 0655 2 !
275 0656 2 ! adjust days for leap year
276 0657 2 ! and make 366 of a non-leap year into a valid date
277 0658 2 !
278 0659 2 LPYEAR = ( .BYEAR [ 0, 0, 2, 0 ] EQLU 0 );
279 0660 2 FEB29 = ( .DAYS EQLU 60 ) AND .LPYEAR;
280 0661 2 IF .LPYEAR
281 0662 2 THEN BEGIN
282 0663 2     IF .DAYS GTRU 59 THEN DAYS = .DAYS - 1;
283 0664 2     END
284 0665 2 ELSE IF .DAYS EQL 366
285 0666 2 THEN
286 0667 2     BEGIN
287 0668 2         YEAR = .YEAR + 1;
288 0669 2         DAYS = 1;
289 0670 2     END;
290 0671 2 !
291 0672 2 ! know days in year and if year is leap year, now is this a legal date
292 0673 2 !
293 0674 2 IF ( .DAYS EQLU 0 ) OR ( .DAYS GTRU 365 ) THEN RETURN 0;
294 0675 2 !
295 0676 2 ! find month
296 0677 2 !
297 0678 2 MONTH = 1;
298 0679 2 WHILE 1
299 0680 2 DO
300 0681 2     BEGIN
301 0682 2         IF ( .DAYS LEQU .DAYTBL[.MONTH] ) THEN EXITLOOP;
302 0683 2         MONTH = .MONTH + 1;
303 0684 2         IF .MONTH GEQU 13 THEN RETURN 0;
304 0685 2     END;
305 0686 2 !
306 0687 2 ! get date in month
307 0688 2 !
308 0689 2 MONTH = .MONTH - 1;
309 0690 2 DAYS = .DAYS - .DAYTBL[.MONTH] + .FEB29;
310 0691 2 !
311 0692 2 ! setup and format output string
312 0693 2 !
313 0694 2 CONVOUTPUT[0] = 11;
314 0695 2 CONVOUTPUT[1] = .DATEREG;
315 0696 2 SYSS$FAO ( CONVCONTROL, 0, CONVOUTPUT, .DAYS, 3, MONTBL[.MONTH], .YEAR );
316 0697 2 RETURN 1;
317 0698 2 !
318 0699 1 END; ! routine
```

4C 5A 34 21 2D 44 41 21 2D 4C 5A 32 21 00131 P.AAI: .ASCII \!2ZL-!AD-!4ZL\

```

00000000D, 0013E .BLKB 2
00000000D, 00140 P.AAH: .LONG 13
00000000D, 00144 .ADDRESS P.AAI

```

CONVCONTROL= P.AAH

[illegible]

```

                                36 11 0009D 7$: BRB 9$
                                50 D7 0009F 8$: DECL MONTH
                                6440 3C 000A1 MOVZWL DAYTBL[MONTH], R3
                                53 C2 000A5 SUBL2 R3, R1
                                52 C1 000A8 ADDL3 FEB29, R1, DAYS
                                0B D0 000AC MOVL #11, CONVOUTPUT
                                04 AC D0 000B0 MOVL DATÉREG, CONVOUTPUT+4
                                04 AE DD 000B5 PUSHL YEAR
                                1A A440 DF 000B8 PUSHAL MONTBL[MONTH]
                                03 DD 000BC PUSHL #3
                                0C AE DD 000BE PUSHL DAYS
                                18 AE 9F 000C1 PUSHAB CONVOUTPUT
                                7E D4 000C4 CLRL -(SP)
                                0138 C4 9F 000C6 PUSHAB CONVCONTROL
                                07 FB 000CA CALLS #7, @#SYSS$FA0
                                01 D0 000D1 MOVL #1, R0
                                50 D4 000D5 9$: RET R0
                                04 000D7 RET

```

; Routine Size: 216 bytes, Routine Base: \$CODE\$ + 0148

```

: 319 0700 1
: 320 0701 1 END
: 321 0702 1
: 322 0703 0 ELUDOM

```

! end module

# PSECT SUMMARY

| Name     | Bytes | Attributes                                                |
|----------|-------|-----------------------------------------------------------|
| \$CODE\$ | 544   | NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2) |

# Library Statistics

| File                            | -----<br>Total | Symbols<br>Loaded | -----<br>Percent | Pages<br>Mapped | Processing<br>Time |
|---------------------------------|----------------|-------------------|------------------|-----------------|--------------------|
| _\$255\$DUA28:[SYSLIB]LIB.L32;1 | 18619          | 4                 | 0                | 1000            | 00:01.8            |

# COMMAND QUALIFIERS

DATECN  
V04-000

B 7  
16-Sep-1984 02:15:00 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:46:37 [MTAACP.SRC]DATECN.B32;1

Page 11  
(3)

```
;      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:DATECN/OBJ=OBJ$:DATECN MSRC$:DATECN/UPDATE=(ENHS:DATECN)
;      323      0704 0
; Size:      417 code + 127 data bytes
; Run Time:      00:11.4
; Elapsed Time:  00:35.0
; Lines/CPU Min: 3711
; Lexemes/CPU-Min: 16555
; Memory Used:  122 pages
; Compilation Complete
```

0254 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY