

RRRRRRRR	PPPPPPPP	GGGGGGGG	UU	UU	DDDDDDDD	AAAAAA	TTTTTTTTTT	EEEEEEEEEE	
RRRRRRRR	PPPPPPPP	GGGGGGGG	UU	UU	DDDDDDDD	AAAAAA	TTTTTTTTTT	EEEEEEEEEE	
RR RR	PP PP	GG	UU	UU	DD DD	AA AA	TT	EE	
RR RR	PP PP	GG	UU	UU	DD DD	AA AA	TT	EE	
RR RR	PP PP	GG	UU	UU	DD DD	AA AA	TT	EE	
RRRRRRRR	PPPPPPPP	GG	UU	UU	DD DD	AA AA	TT	EEEEEEEEEE	
RRRRRRRR	PPPPPPPP	GG	UU	UU	DD DD	AA AA	TT	EEEEEEEEEE	
RR RR	PP	GG GGGGGG	UU	UU	DD DD	AAAAAAAAAA	TT	EE	
RR RR	PP	GG GGGGGG	UU	UU	DD DD	AAAAAAAAAA	TT	EE	
RR RR	PP	GG GG	UU	UU	DD DD	AA AA	TT	EE
RR RR	PP	GG GG	UU	UU	DD DD	AA AA	TT	EE
RR RR	PP	GGGGGG	UUUUUUUUUU	UUUUUUUUUU	DDDDDDDD	AA AA	TT	EEEEEEEEEE
RR RR	PP	GGGGGG	UUUUUUUUUU	UUUUUUUUUU	DDDDDDDD	AA AA	TT	EEEEEEEEEE

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	SS
LL	II	SS	SS
LL	II	SS	SS
LL	II	SS	SS
LLLLLLLLLL	IIIIII	SSSSSSSS	
LLLLLLLLLL	IIIIII	SSSSSSSS	

```

1 0001 0 MODULE RPG$UPDATE( %TITLE 'Support UDATE, UDAY, UMONTH, UYEAR special words'
2 0002 0 IDENT = '1-002' ! file: RPGUPDATE.B32 EDIT:DG1002
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1
31 0031 1 +-
32 0032 1 FACILITY: RPGII SUPPORT
33 0033 1
34 0034 1 ABSTRACT
35 0035 1
36 0036 1 This routine supports the RPG UDATE, UDAY, UMONTH and UYEAR
37 0037 1 special words.
38 0038 1
39 0039 1
40 0040 1 ENVIRONMENT: Vax-11 User Mode
41 0041 1
42 0042 1 AUTHOR: Debess Grabazs, CREATION DATE: 25-JAN-1983
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 1-001 - Original. DG 25-JAN-1983
47 0047 1 1-002 - Take out %ASCID - was making code unsharable. DG 11-Jul-1983
48 0048 1 --
49 0049 1
50 0050 1 !<BLF/PAGE>

```

```

: 52 0051 1 %SBTTL 'Declarations'
: 53 0052 1 | +
: 54 0053 1 | PROLOGUE FILE:
: 55 0054 1 | -
: 56 0055 1
: 57 0056 1 REQUIRE 'RTLIN:RPGPROLOG';           ! Switches, PSECTs, macros,
: 58 0121 1                                     ! linkages and LIBRARYs
: 59 0122 1
: 60 0123 1 | +
: 61 0124 1 | LINKAGES
: 62 0125 1 |     NONE
: 63 0126 1 | -
: 64 0127 1
: 65 0128 1 | +
: 66 0129 1 | TABLE OF CONTENTS:
: 67 0130 1 | -
: 68 0131 1
: 69 0132 1 FORWARD ROUTINE
: 70 0133 1     RPG$UPDATE : NOVALUE ;
: 71 0134 1
: 72 0135 1 | +
: 73 0136 1 | INCLUDE FILES
: 74 0137 1 |     NONE
: 75 0138 1 | -
: 76 0139 1
: 77 0140 1 | +
: 78 0141 1 | MACROS
: 79 0142 1 |     NONE
: 80 0143 1 | -
: 81 0144 1
: 82 0145 1 | +
: 83 0146 1 | EQUATED SYMBOLS
: 84 0147 1 |     NONE
: 85 0148 1 | -
: 86 0149 1
: 87 0150 1 | +
: 88 0151 1 | EXTERNAL REFERENCES
: 89 0152 1 | -
: 90 0153 1
: 91 0154 1 EXTERNAL ROUTINE
: 92 0155 1     COB$ACC DATE,           ! Get the current date
: 93 0156 1     LIB$STOP,             ! Stop execution via signalling
: 94 0157 1     STR$UPCASE;           ! Convert string to upper case
: 95 0158 1
: 96 0159 1 EXTERNAL LITERAL
: 97 0160 1     RPG$_FATINTERR,       ! Fatal internal error
: 98 0161 1     RPG$_INVUDATE;       ! Incorrectly specified date

```

```

100 0162 1 %SBTTL 'RPG$UPDATE - Support UDATE, UDAY, UMONTH, UYEAR special words'
101 0163 1 GLOBAL ROUTINE RPG$UPDATE(
102 0164 1     DAY,           ! Current day
103 0165 1     MONTH,        ! Current month
104 0166 1     YEAR,         ! Current year
105 0167 1 ): NOVALUE=
106 0168 1
107 0169 1
108 0170 1 ++
109 0171 1
110 0172 1 FUNCTIONAL DESCRIPTION
111 0173 1
112 0174 1     This routine translates the logical name RPG$UPDATE. If
113 0175 1     the logical translates, the date specified in the logical
114 0176 1     name is used; otherwise the current system date is used.
115 0177 1     The format of the date in RPG$UPDATE is dd-mmm-yyyy.
116 0178 1     Two character strings representing the day, numeric month
117 0179 1     and last two digits of the year are moved into the appropriate
118 0180 1     return parameters.
119 0181 1     This routine is called when an RPGII program has any of the
120 0182 1     special words UDATE, UDAY, UMONTH, or UYEAR in it.
121 0183 1
122 0184 1 CALLING SEQUENCE:
123 0185 1
124 0186 1     CALL RPG$UPDATE (day.wt.r, month.wt.r, year.wt.r)
125 0187 1
126 0188 1 FORMAL PARAMETERS:
127 0189 1
128 0190 1     day           address of 2 character string to receive
129 0191 1                 current day of the month
130 0192 1
131 0193 1     month        address of 2 character string to receive
132 0194 1                 current month number
133 0195 1
134 0196 1     year         address of 2 character string to receive
135 0197 1                 current year
136 0198 1
137 0199 1
138 0200 1 IMPLICIT INPUTS:
139 0201 1
140 0202 1     RPG$UPDATE -  logical name which may have the date
141 0203 1                 specified in it
142 0204 1
143 0205 1 IMPLICIT OUTPUTS:
144 0206 1
145 0207 1     NONE
146 0208 1
147 0209 1 COMPLETION CODES:
148 0210 1
149 0211 1     NONE
150 0212 1
151 0213 1 SIDE EFFECTS:
152 0214 1
153 0215 1     Errors will be signalled if the date specified in RPG$UPDATE
154 0216 1     is incorrectly specified or if there is an error in the
155 0217 1     translation of the logical.
156 0218 1

```

RPG\$UPDATE
1-002

Support: UDATE, UDAY, UMONTH, UYEAR special word 16-Sep-1984 02:19:50
RPG\$UPDATE - Support UDATE, UDAY, UMONTH, UYEAR 14-Sep-1984 13:04:26

E 5

VAX-11 Bliss-32 V4.0-742
[RPGRTL.SRC]RPGUPDATE.B32;1

Page 4
(3)

: 157
: 158
: 159
: 160

0219 1 !--
0220 1 !
0221 1 !
0222 1 !<BLF/PAGE>

RPC
Sym
RPC
RPC
RPC
RPC
RPC
RPC
RPC
RPC
RPC
RPC
RPC

PSE

\$RF

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Crc
Ass

The
222
The
141
1 p

Mac

_Si
O ()
The
MA

```

162 0223 1
163 0224 2 BEGIN
164 0225 2
165 0226 2 LOCAL
166 0227 2 DASH_PTR, ! Pointer to 1st dash
167 0228 2 DEST_DESC: BLOCK [8, BYTE], ! Local desc for call to $TRNLOG
168 0229 2 DIFF, ! Size of year part
169 0230 2 END_PTR, ! Pointer past end of logical name string
170 0231 2 LOG_DESC: BLOCK [8, BYTE], ! Local desc for call to $TRNLOG
171 0232 2 MONTH_FOUND: BYTE INITIAL(0), ! Indicator of valid month
172 0233 2 MONTH_TEXT: VECTOR [3, BYTE], ! Local string for uppercase month
173 0234 2 NAME_LENGTH, ! Length of logical name string
174 0235 2 NAME_TEXT: VECTOR [LNMSC_NAMLENGTH, BYTE], ! Local string for logical name
175 0236 2 RET_STATUS; ! Return status
176 0237 2
177 0238 2 BIND
178 0239 2 MONTHNAME = UPLIT(
179 0240 2 'JANFEBMARAPR MAYJUNJUL AUGSEP OCTNOVDEC'), ! Month names
180 0241 2
181 0242 2 MONTHNUM = UPLIT(
182 0243 2 '010203040506070809101112'), ! Month numbers
183 0244 2
184 0245 2 P_DASH = UPLIT ('-');
185 0246 2 P_ZERO = UPLIT ('0');
186 0247 2
187 0248 2 !+
188 0249 2 ! Initialize descriptors for call to $TRNLOG.
189 0250 2
190 0251 2 !-
191 0252 2 DEST_DESC[DSC$W_LENGTH] = LNMSC_NAMLENGTH;
192 0253 2 DEST_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
193 0254 2 DEST_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
194 0255 2 DEST_DESC[DSC$A_POINTER] = NAME_TEXT;
195 0256 2
196 0257 2 LOG_DESC[DSC$W_LENGTH] = %CHARCOUNT ('RPG$UPDATE');
197 0258 2 LOG_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
198 0259 2 LOG_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
199 0260 2 LOG_DESC[DSC$A_POINTER] = UPLIT ('RPG$UPDATE');
200 0261 2
201 0262 2 !+
202 0263 2 ! Try to translate the logical name RPG$UPDATE.
203 0264 2
204 0265 2 !-
205 0266 2 RET_STATUS = $TRNLOG (LOGNAM = LOG_DESC, RSLLEN = NAME_LENGTH, RSLBUF = DEST_DESC);
206 0267 2
207 0268 2 IF .RET_STATUS EQL SS$NORMAL
208 0269 2 THEN
209 0270 2 !+
210 0271 2 ! Logical name translated - use date specified ('dd-mmm-yyyy').
211 0272 2
212 0273 2 !-
213 0274 2 BEGIN ! Logical translated
214 0275 2
215 0276 2 IF CH$FAIL (DASH_PTR = CH$FIND_CH (
216 0277 2 2, ! Find first dash in date
217 0278 2 ! Should be in 2nd or 3rd
218 0279 2

```

```

219 0280 3 .DEST_DESC[DSC$A_POINTER]+1, ! position only
220 0281 3 %C('-')
221 0282 3
222 0283 3 THEN
223 0284 3 LIB$STOP (RPG$_INVUDATE)
224 0285 3 ELSE
225 0286 3 +
226 0287 3 Found first dash, now process day part of date.
227 0288 3
228 0289 3 -
229 0290 4 BEGIN ! Found 1st dash
230 0291 4
231 0292 4 LOCAL
232 0293 4 DEST2_DESC: BLOCK [8, BYTE]; ! Local desc for call to STR$UPCASE
233 0294 4
234 0295 4 IF CH$DIFF (.DASH_PTR, .DEST_DESC[DSC$A_POINTER]) EQL %CHARCOUNT ('d')
235 0296 4 THEN
236 0297 4 +
237 0298 4
238 0299 4 Day part is only one character,
239 0300 4 so must pad with a leading zero.
240 0301 4
241 0302 4 -
242 0303 5 BEGIN
243 0304 5
244 0305 5 CH$MOVE (1, P_ZERO, .DAY);
245 0306 5 CH$MOVE (1, .DEST_DESC[DSC$A_POINTER], .DAY+1);
246 0307 5
247 0308 5 END
248 0309 4 ELSE
249 0310 4 CH$MOVE (2, .DEST_DESC[DSC$A_POINTER], .DAY);
250 0311 4
251 0312 4 +
252 0313 4 Process year part of date.
253 0314 4 DASH_PTR still points to 1st dash (-mmm-yyyy).
254 0315 4
255 0316 4 -
256 0317 4 IF CH$EQL (1, P_DASH, 1, .DASH_PTR+4)
257 0318 4 THEN
258 0319 4 +
259 0320 4
260 0321 4 Found second dash, now determine how
261 0322 4 many characters in year part of date.
262 0323 4
263 0324 4 -
264 0325 5 BEGIN ! Found 2nd dash
265 0326 5
266 0327 5 END_PTR = .DEST_DESC[DSC$A_POINTER] + .NAME_LENGTH;
267 0328 5 DIFF = CH$DIFF (.END_PTR, .DASH_PTR+5);
268 0329 5 IF .DIFF GTR %CHARCOUNT ('y')
269 0330 5 THEN
270 0331 5 +
271 0332 5
272 0333 5 Correctly specified year part (at least 2 chars).
273 0334 5
274 0335 5 -
275 0336 5 CH$MOVE (2, .END_PTR-2, .YEAR)

```


276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332

0337 5
0338 5
0339 5
0340 6
0341 6
0342 6
0343 6
0344 6
0345 6
0346 6
0347 6
0348 6
0349 6
0350 6
0351 5
0352 5
0353 5
0354 5
0355 5
0356 4
0357 4
0358 4
0359 4
0360 4
0361 4
0362 4
0363 4
0364 4
0365 4
0366 4
0367 4
0368 4
0369 4
0370 4
0371 4
0372 4
0373 4
0374 5
0375 4
0376 4
0377 4
0378 4
0379 5
0380 5
0381 5
0382 5
0383 5
0384 5
0385 5
0386 5
0387 5
0388 6
0389 6
0390 6
0391 6
0392 6
0393 6

```
ELSE
  IF .DIFF EQL %CHARCOUNT ('y')
  THEN
    BEGIN
      +
      Year part is only one character,
      so must pad with leading zero.
      -
      CH$MOVE (1, P_ZERO, .YEAR);
      CH$MOVE (1, .END_PTR-1, .YEAR+1);
    END
  ELSE
    LIB$STOP (RPG$_INVUDATE);
END
! Found 2nd dash
ELSE
  LIB$STOP (RPG$_INVUDATE);
+
Process month part of date.
DASH_PTR still points to 1st dash (-mmm-yyyy).
-
DEST_DESC[DSC$W_LENGTH] = %CHARCOUNT ('mmm');
DEST_DESC[DSC$A_POINTER] = .DASH_PTR + 1;
DEST2_DESC[DSC$W_LENGTH] = %CHARCOUNT ('mmm');
DEST2_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
DEST2_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
DEST2_DESC[DSC$A_POINTER] = MONTH_TEXT;
+
Change month specified to uppercase for matching.
IF NOT (STR$UPCASE (DEST2_DESC, DEST_DESC))
THEN
  LIB$STOP (RPG$_FATINTERR);
INCR I FROM 0 TO 11 DO
  BEGIN
    +
    Search for character match in table
    and convert to numerical equivalent.
    -
    IF CH$EQL (3, CH$PTR (MONTHNAME, .I*3), 3, .DEST2_DESC[DSC$A_POINTER])
  THEN
    BEGIN
      CH$MOVE (2, CH$PTR (MONTHNUM, .I*2), .MONTH);
      MONTH_FOUND = 1;
      EXITLOOP;
```

EXE
Moc
CTE
CTE
CTE
SYS
SYS

```

333 0394 5      END;
334 0395 5
335 0396 4      END;
336 0397 4
337 0398 4
338 0399 4      Make sure a match was found.
339 0400 4
340 0401 4
341 0402 4      IF NOT .MONTH_FOUND
342 0403 4      THEN
343 0404 4          LIB$STOP (RPG$_INVUDATE);
344 0405 4
345 0406 4      END;
346 0407 3      ! Found 1st dash
347 0408 3      ! Logical translated
348 0409 2
349 0410 2
350 0411 2
351 0412 2      ELSE
352 0413 2          Logical name did not translate - use current system date.
353 0414 2
354 0415 2      BEGIN
355 0416 3
356 0417 3      IF .RET_STATUS EQL SSS_NOTRAN
357 0418 3      THEN
358 0419 4          BEGIN
359 0420 4
360 0421 4          COB$ACC_DATE (DEST_DESC);
361 0422 4          CH$MOVE (XCHARCOUNT ('yy'), .DEST_DESC[DSC$_A_POINTER], .YEAR);
362 0423 4          CH$MOVE (XCHARCOUNT ('mm'), .DEST_DESC[DSC$_A_POINTER]+2, .MONTH);
363 0424 4          CH$MOVE (XCHARCOUNT ('dd'), .DEST_DESC[DSC$_A_POINTER]+4, .DAY);
364 0425 4
365 0426 4      END
366 0427 3      ELSE
367 0428 3          LIB$STOP (.RET_STATUS);
368 0429 3
369 0430 2
370 0431 2
371 0432 1      END;

```

```

.TITLE  RPGSUDATE Support UDATE, UDAY, UMONTH, UYEAR sp
.IDENT  \1-002\
.PSECT  _RPG$CODE,NOWRT, SHR, PIC,2
59 41 4D 52 50 41 52 41 4D 42 45 46 4E 41 4A 00000 P.AAA: .ASCII \JANFEBMARAPR MAYJUNJULAUGSEPOCTNOVDEC\
54 43 4F 50 45 53 47 55 41 4C 55 4A 4E 55 4A 0000F
43 45 44 56 4F 4E 0001E
30 37 30 36 30 35 30 34 30 33 30 32 30 31 30 00024 P.AAB: .ASCII \010203040506070809101112\
32 31 31 31 30 31 39 30 38 00033
00 00 00 2D 0003C P.AAC: .ASCII \-\<0><0><0>
00 00 00 30 00040 P.AAD: .ASCII \0\<0><0><0>
00 00 00 45 54 41 44 55 24 47 50 52 00044 P.AAE: .ASCII \RPGSUDATE\<0><0><0>
MONTHNAME= P.AAA

```

MONTHNUM=	P.AAB	COBSACC DATE, LIB\$STOP	
P_DASH=	P.AAC	STR\$UPCASE, RPG\$ FATINTERR	
P_ZERO=	P.AAD	RPG\$ INVUDATE, SYS\$TRNLOG	
	.EXTRN	RP\$UPDATE, Save R2,R3,R4,R5,R6,R7,R8	0163
	MOVAB	#RPG\$ INVUDATE, R8	
	MOVAB	P_ZERO, R7	
	MOVAB	LIB\$STOP, R6	
	MOVAB	-288(SP), SP	
	CLRB	MONTH_FOUND	0224
F8 AD	MOVAB	#17694975, DEST_DESC	0253
FC AD	MOVAB	NAME_TEXT, DEST_DESC+4	0256
F0 AD	MOVAB	#17694729, LOG_DESC	0258
F4 AD	MOVAB	P.AAE, LOG_DESC+4	0261
	CLRB	-(SP)	0268
	CLRL	-(SP)	
	PUSHAB	DEST_DESC	
	PUSHAB	NAME_LENGTH	
	PUSHAB	LOG_DESC	
	CALLS	#6, SYS\$TRNLOG	
	CPL	RET_STATUS, #1	0269
	BEQL	1\$	
	BRW	15\$	
	MOVAB	DEST_DESC+4, R2	0280
01 A2	LOCC	#45, #2, 1(R2)	0278
	BNEQ	2\$	
	CLRL	R1	
	MOVAB	R1, DASH_PTR	
	BNEQ	4\$	0281
	PUSHL	R8	0283
	BRW	17\$	
	MOVAB	1(R2), R0	0295
	CPL	DASH_PTR, R0	
	BNEQ	5\$	
	MOVAB	P_ZERO, @DAY	0305
	MOVAB	DAY, R0	
	MOVAB	(R2), 1(R0)	0306
	BRB	6\$	0295
	MOVW	(R2), @DAY	0310
	CMPB	P_DASH, 4(DASH_PTR)	0317
	BNEQ	8\$	
	ADDL3	NAME_LENGTH, R2, END_PTR	0327
	SUBL3	DASH_PTR, END_PTR, RT	0328
	SUBL2	#5, DIFF	
	CPL	DIFF, #1	0329
	BLEQ	7\$	
	MOVW	-2(END_PTR), @YEAR	0336
	BRB	9\$	
	BNEQ	8\$	0338
	MOVAB	P_ZERO, @YEAR	0347
	MOVAB	YEAR, R1	
	MOVAB	-1(END_PTR), 1(R1)	0348
	BRB	9\$	0338
	PUSHL	R8	0357

FB	AD		01	FB	000B4	CALLS	#1, LIB\$STOP		
FC	AD		03	B0	000B7	MOVW	#3, DEST_DESC	9\$:	0365
08	AE	010E0003	A3	9E	000BB	MOVAB	1(R3), DEST_DESC+4		0366
OC	AE		8F	D0	000C0	MOVL	#17694723, DEST2_DESC		0367
			04	AE	000C8	MOVAB	MONTH_TEXT, DEST2_DESC+4		0370
			F8	AD	000CD	PUSHAB	DEST_DESC		0374
			OC	AE	000D0	PUSHAB	DEST2_DESC		
00000000G	00		02	FB	000D3	CALLS	#2, STR\$UPCASE		
	09		50	E8	000DA	BLBS	RO, 10\$		
	66	00000000G	8F	DD	000DD	PUSHL	#RPG\$ FATINTERR		0376
			01	FB	000E3	CALLS	#1, LIB\$STOP		
			54	D4	000E6	CLRL	I	10\$:	0386
OC	50		03	C5	000E8	MULL3	#3, I, RO	11\$:	
BE	CO	A740	03	29	000EC	CMFC3	#3, MONTHNAME[RO], @DEST2_DESC+4		
			0B	12	000F3	BNEQ	12\$		
	08		E4	A744	B0	000F5	MOVW	MONTHNUM[I], @MONTH	0390
					01	90	MOVW	#1, MONTH_FOUND	0391
					04	11	BRB	13\$	0388
	E4				0B	F3	AOBLEQ	#11, I, 11\$	0378
					55	E8	BLBS	MONTH_FOUND, 14\$	0402
					FF59	31	BRW	3\$	
					04	0010A	RET		0404
00000629	8F			50	D1	0010B	CMP	RET_STATUS, #1577	0417
				1D	12	00112	BNEQ	16\$	
				F8	AD	9F	PUSHAB	DEST_DESC	0421
00000000G	00			01	FB	00117	CALLS	#1, COB\$ACC_DATE	
	50			FC	AD	D0	MOVL	DEST_DESC+4, RO	0422
	OC			60	B0	00122	MOVW	(RO), @YEAR	
	08			02	A0	B0	MOVW	2(RO), @MONTH	0423
	04			04	A0	B0	MOVW	4(RO), @DAY	0424
					04	00130	RL		0417
				50	DD	00131	PUSHL	RET_STATUS	0428
	66			01	FB	00133	CALLS	#1, LIB\$STOP	
				04	00136		RET		0432

; Routine Size: 311 bytes, Routine Base: _RPG\$CODE + 0050

: 372 0433 1
: 373 0434 0 END ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
_RPG\$CODE	391	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	12	0	581	00:01.0
\$_\$255\$DUA28:[RPGRTL.OBJ]RPGLIB.L32;1	54	0	0	9	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:RPGUPDATE/OBJ=OBJ\$:RPGUPDATE MSRCS\$:RPGUPDATE/UPDATE=(ENHS\$:RPGUPDATE)

: Size: 311 code + 80 data bytes
 : Run Time: 00:08.5
 : Elapsed Time: 00:27.1
 : Lines/CPU Min: 3059
 : Lexemes/CPU-Min: 14044
 : Memory Used: 130 pages
 : Compilation Complete

