

[illegible]

RPC
Tal

[illegible]


```
0001 0 MODULE RPG$UPDATE( %TITLE 'Support UDATE, UDAY, UMONTH, UYEAR special words'
0002 0 IDENT = '1-002' ! file: RPGUPDATE.B32 EDIT:DG1002
0003 0 ) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *
0027 1 *****
0028 1
0029 1
0030 1
0031 1 ++
0032 1 FACILITY:      RPGII SUPPORT
0033 1
0034 1 ABSTRACT
0035 1
0036 1 This routine supports the RPG UDATE, UDAY, UMONTH and UYEAR
0037 1 special words.
0038 1
0039 1
0040 1 ENVIRONMENT:  Vax-11 User Mode
0041 1
0042 1 AUTHOR: Debess Grabazs, CREATION DATE: 25-JAN-1983
0043 1
0044 1 MODIFIED BY:
0045 1
0046 1 1-001 - Original. DG 25-JAN-1983
0047 1 1-002 - Take out %ASCID - was making code unsharable. DG 11-Jul-1983
0048 1 --
0049 1
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';
58 0121 1 | Switches, PSECTs, macros,
59 0122 1 | linkages and LIBRARYs
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,
93 0156 1 LIB$STOP,
94 0157 1 STR$UPCASE;
95 0158 1
96 0159 1 EXTERNAL LITERAL
97 0160 1 RPG$_FATINTERR,
98 0161 1 RPG$_INVUDATE;

! Get the current date
! Stop execution via signalling
! Convert string to upper case

! Fatal internal error
! 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

: 157
: 158
: 159
: 160

Support UPDATE, UDAY, UMONTH, UYEAR special word
RPG\$UPDATE - Support UPDATE, UDAY, UMONTH, UYEAR

E 5

16-Sep-1984 02:19:50
14-Sep-1984 13:04:26

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

Page 4
(3)

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

RPG
Syn
RPG
RPG
RPG
RPG
RPG
RPG
RPG
RPG
RPG
RPG
RPG

PSE

\$RF

Pha

Ini
Com
Pas
Syn
Pas
Syn
Pse
Cro
Ass

The
222
The
141
1

Mac

_S
0
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 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC'), ! 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$A_POINTER]+1, ! position only
220 0281 3 %C'-')
221 0282 3 THEN
222 0283 3 LIB$STOP (RPG$_INVDATE)
223 0284 3 ELSE
224 0285 3 +
225 0286 3 Found first dash, now process day part of date.
226 0287 3 -
227 0288 3 BEGIN ! Found 1st dash
228 0289 3
229 0290 4
230 0291 4
231 0292 4 LOCAL
232 0293 4 DEST2_DESC: BLOCK [8, BYTE]; ! Local desc for call to STR$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 Day part is only one character,
238 0299 4 so must pad with a leading zero.
239 0300 4 -
240 0301 4 BEGIN
241 0302 4
242 0303 5 CH$MOVE (1, P_ZERO, .DAY);
243 0304 5 CH$MOVE (1, .DEST_DESC[DSC$A_POINTER], .DAY+1);
244 0305 5
245 0306 5 END
246 0307 5 ELSE
247 0308 5 CH$MOVE (2, .DEST_DESC[DSC$A_POINTER], .DAY);
248 0309 5 +
249 0310 5 Process year part of date.
250 0311 5 DASH_PTR still points to 1st dash (-mmm-yyyy).
251 0312 5 -
252 0313 5 IF CH$EQL (1, P_DASH, 1, .DASH_PTR+4)
253 0314 5 THEN
254 0315 5 +
255 0316 5 Found second dash, now determine how
256 0317 5 many characters in year part of date.
257 0318 5 -
258 0319 5 BEGIN ! Found 2nd dash
259 0320 5
260 0321 5 END_PTR = .DEST_DESC[DSC$A_POINTER] + .NAME_LENGTH;
261 0322 5 DIFF = CH$DIFF (.END_PTR, .DASH_PTR+5);
262 0323 5 IF .DIFF GTR %CHARCOUNT ('y')
263 0324 5 THEN
264 0325 5 +
265 0326 5 Correctly specified year part (at least 2 chars).
266 0327 5 -
267 0328 5 CH$MOVE (2, .END_PTR-2, .YEAR)
268 0329 5
269 0330 5
270 0331 5
271 0332 5
272 0333 5
273 0334 5
274 0335 5
275 0336 5
```



```
276      0337 5      ELSE
277      0338 5      IF .DIFF EQL %CHARCOUNT ('y')
278      0339 5      THEN
279      0340 6      BEGIN
280      0341 6      +
281      0342 6      |
282      0343 6      |   Year part is only one character,
283      0344 6      |   so must pad with leading zero.
284      0345 6      |
285      0346 6      |
286      0347 6      |   CH$MOVE (1, P_ZERO, .YEAR);
287      0348 6      |   CH$MOVE (1, .END_PTR-1, .YEAR+1);
288      0349 6      |
289      0350 6      |
290      0351 5      END
291      0352 5      ELSE
292      0353 5      LIB$STOP (RPG$_INVUDATE);
293      0354 5      END
294      0355 5      ! Found 2nd dash
295      0356 4      ELSE
296      0357 4      LIB$STOP (RPG$_INVUDATE);
297      0358 4
298      0359 4      +
299      0360 4      |
300      0361 4      |   Process month part of date.
301      0362 4      |   DASH_PTR still points to 1st dash (-mmm-yyyy).
302      0363 4      |
303      0364 4      |
304      0365 4      |   DEST_DESC[DSC$_LENGTH] = %CHARCOUNT ('mmm');
305      0366 4      |   DEST_DESC[DSC$_POINTER] = .DASH_PTR + 1;
306      0367 4      |   DEST2_DESC[DSC$_LENGTH] = %CHARCOUNT ('mmm');
307      0368 4      |   DEST2_DESC[DSC$_CLASS] = DSC$_CLASS_S;
308      0369 4      |   DEST2_DESC[DSC$_DTYPE] = DSC$_DTYPE_T;
309      0370 4      |   DEST2_DESC[DSC$_POINTER] = MONTH_TEXT;
310      0371 4      |
311      0372 4      |   +
312      0373 4      |   |   Change month specified to uppercase for matching.
313      0374 5      |   IF NOT (STR$UPCASE (DEST2_DESC, DEST_DESC))
314      0375 4      |   THEN
315      0376 4      |   |   LIB$STOP (RPG$_FATINTERR);
316      0377 4      |
317      0378 4      |   INCR I FROM 0 TO 11 DO
318      0379 5      |   |   BEGIN
319      0380 5      |   |   +
320      0381 5      |   |   |
321      0382 5      |   |   |   Search for character match in table
322      0383 5      |   |   |   and convert to numerical equivalent.
323      0384 5      |   |   |
324      0385 5      |   |   |
325      0386 5      |   |   |   IF CH$EQL (3, CH$PTR (MONTHNAME, .I*3), 3, .DEST2_DESC[DSC$_POINTER])
326      0387 5      |   |   |   THEN
327      0388 6      |   |   |   |   BEGIN
328      0389 6      |   |   |   |   |
329      0390 6      |   |   |   |   |   CH$MOVE (2, CH$PTR (MONTHNUM, .I*2), .MONTH);
330      0391 6      |   |   |   |   |   MONTH_FOUND = 1;
331      0392 6      |   |   |   |   |   EXITLOOP;
332      0393 6      |   |   |   |   |
```

RPGSUPDATE
1-002

Support UDATE, UDAY, UMONTH, UYEAR special word
RPGSUPDATE - Support UDATE, UDAY, UMONTH, UYEAR

1 5
16-Sep-1984 02:19:50
14-Sep-1984 13:04:26

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

Page 8
(4)

```

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$STOP (RPG$_INVUPDATE);
344      0405 4
345      0406 3      END;
346      0407 3
347      0408 3      END
348      0409 2      ELSE
349      0410 2      +
350      0411 2      -
351      0412 2      Logical name did not translate - use current system date.
352      0413 2
353      0414 2      -
354      0415 3      BEGIN
355      0416 3
356      0417 3      IF .RET_STATUS EQL SS$_NOTRAN
357      0418 3      THEN
358      0419 4          BEGIN
359      0420 4
360      0421 4          COB$ACC_DATE (DEST_DESC);
361      0422 4          CH$MOVE (%CHARCOUNT ('yy'), .DEST_DESC[DSC$_POINTER], .YEAR);
362      0423 4          CH$MOVE (%CHARCOUNT ('mm'), .DEST_DESC[DSC$_POINTER]+2, .MONTH);
363      0424 4          CH$MOVE (%CHARCOUNT ('dd'), .DEST_DESC[DSC$_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      END;
370      0431 2
371      0432 1      END;

```

.TITLE RPG\$UPDATE Support UDATE, UDAY, UMONTH, UYEAR sp
ecial word

.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
30 37 30 36 30 35 30 34 30 33 45 44 56 4F 4E 0001E
32 31 31 31 30 32 30 31 30 31 39 30 38 00024 P.AAB: .ASCII \010203040506070809101112\
00 00 00 00 00 00 00 00 2D 0003C P.AAC: .ASCII \-\<0><0><0>
00 00 00 00 00 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 \RPG$UPDATE\<0><0><0>

```

MONTHNAME=

P.AAA

				MONTHNUM=	P.AAB			
				P_DASH=	P.AAC			
				P_ZERO=	P.AAD			
				.EXTRN	COB\$ACC DATE, LIB\$STOP			
				.EXTRN	STR\$UPCASE, RPG\$ FATINTERR			
				.EXTRN	RPG\$ INVUDATE, SYS\$TRNLOG			
				01FC	00000	.ENTRY	RPG\$UPDATE, Save R2,R3,R4,R5,R6,R7,R8	0163
	58	00000000G	8F	D0	00002	MOVL	#RPG\$ INVUDATE, R8	
	57	E4	AF	9E	00009	MOVAB	P_ZERO, R7	
	56	00000000G	00	9E	0000D	MOVAB	LIB\$STOP, R6	
	5E	FEE0	CE	9E	00014	MOVAB	-288(SP), SP	
			55	94	00019	CLRB	MONTH FOUND	0224
F8	AD	010E00FF	8F	D0	0001B	MOVL	#17694975, DEST_DESC	0253
FC	AD	10	AE	9E	00023	MOVAB	NAME TEXT, DEST_DESC+4	0256
F0	AD	010E0009	8F	D0	00028	MOVL	#17694729, LOG_DESC	0258
F4	AD	04	A7	9E	00030	MOVAB	P.AAE, LOG_DESC+4	0261
			7E	7C	00035	CLRQ	-(SP)	0268
			7E	D4	00037	CLRL	-(SP)	
			F8	AD	9F	PUSHAB	DEST_DESC	
			10	AE	9F	PUSHAB	NAME_LENGTH	
			F0	AD	9F	PUSHAB	LOG_DESC	
	00000000G	00	06	FB	00042	CALLS	#6, SYS\$TRNLOG	
		01	50	D1	00049	CMPL	RET_STATUS, #1	0269
			03	13	0004C	BEQL	1\$	
			00BA	31	0004E	BRW	15\$	
		52	AD	D0	00051	MOVL	DEST_DESC+4, R2	0280
01	A2	02	2D	3A	00055	LOCC	#45, #2, 1(R2)	0278
			02	12	0005A	BNEQ	2\$	
			51	D4	0005C	CLRL	R1	
		53	51	D0	0005E	MOVL	R1, DASH_PTR	
			05	12	00061	BNEQ	4\$	0281
			58	DD	00063	PUSHL	R8	0283
			00CB	31	00065	BRW	17\$	
		50	A2	9E	00068	MOVAB	1(R2), R0	0295
		50	53	D1	0006C	CMPL	DASH_PTR, R0	
			0E	12	0006F	BNEQ	5\$	
04	BC		67	90	00071	MOVB	P_ZERO, @DAY	0305
			AC	D0	00075	MOVL	DAY, R0	
01	A0		62	90	00079	MOVB	(R2), 1(R0)	0306
			04	11	0007D	BRB	6\$	0295
04	BC		62	B0	0007F	MOVW	(R2), @DAY	0310
04	A3		A7	91	00083	CMPB	P_DASH, 4(DASH_PTR)	0317
			28	12	00088	BNEQ	8\$	
		52	6E	C1	0008A	ADDL3	NAME_LENGTH, R2, END_PTR	0327
		50	53	C3	0008E	SUBL3	DASH_PTR, END_PTR, RT	0328
		51	05	C2	00092	SUBL2	#5, DIFF	
		01	51	D1	00095	CMPL	DIFF, #1	0329
			07	15	00098	BLEQ	7\$	
0C	BC		A0	B0	0009A	MOVW	-2(END_PTR), @YEAR	0336
			16	11	0009F	BRB	9\$	
			0F	12	000A1	BNEQ	8\$	0338
			67	90	000A3	MOVB	P_ZERO, @YEAR	0347
0C	BC		AC	D0	000A7	MOVL	YEAR, R1	
			A0	90	000AB	MOVB	-1(END_PTR), 1(R1)	0348
01	A1		05	11	000B0	BRB	9\$	0338
			58	DD	000B2	PUSHL	R8	0357

			66	01	FB	000B4	CALLS	#1, LIB\$STOP			
	F8	AD		03	B0	000B7	9\$: MOVW	#3, DEST_DESC	0365		
	FC	AD	01	A3	9E	000BB	MOVAB	1(R3), DEST_DESC+4	0366		
	08	AE	010E0003	8F	D0	000C0	MOVL	#17694723, DEST2_DESC	0367		
	OC	AE		04	AE	9E	000C8	MOVAB	MONTH_TEXT, DEST2_DESC+4	0370	
				F8	AD	9F	000CD	PUSHAB	DEST_DESC	0374	
				OC	AE	9F	000D0	PUSHAB	DEST2_DESC		
	00000000G	00		02	FB	000D3	CALLS	#2, STR\$UPCASE			
		09		50	E8	000DA	BLBS	R0, 10\$			
			00000000G	8F	DD	000DD	PUSHL	#RPG\$FATINTERR	0376		
		66		01	FB	000E3	CALLS	#1, LIB\$STOP			
				54	D4	000E6	10\$: CLRL	I	0386		
	50	54		03	C5	000E8	11\$: MULL3	#3, I, R0			
OC	BE	CO	A740	03	29	000EC	CMFC3	#3, MONTHNAME[R0], @DEST2_DESC+4			
				0B	12	000F3	BNEQ	12\$			
	08	BC	E4	A744	B0	000F5	MOVW	MONTHNUM[I], @MONTH	0390		
		55		01	90	000FB	MOVB	#1, MONTH_FOUND	0391		
				04	11	000FE	BRB	13\$	0388		
	E4	54		0B	F3	00100	12\$: AOBLEQ	#11, I, 11\$	0378		
		03		55	E8	00104	13\$: BLBS	MONTH_FOUND, 14\$	0402		
				FF59	31	00107	BRW	3\$			
					04	0010A	14\$: RET		0404		
	00000629	8F		50	D1	0010B	15\$: CMPL	RET_STATUS, #1577	0417		
				1D	12	00112	BNEQ	16\$			
				F8	AD	9F	00114	PUSHAB	DEST_DESC	0421	
	00000000G	00		01	FB	00117	CALLS	#1, COB\$ACC_DATE			
		50		FC	AD	D0	0011E	MOVL	DEST_DESC+4, R0	0422	
	OC	BC		60	B0	00122	MOVW	(R0), @YEAR			
	08	BC	02	A0	B0	00126	MOVW	2(R0), @MONTH	0423		
	04	BC	04	A0	B0	0012B	MOVW	4(R0), @DAY	0424		
					04	00130	RET		0417		
				50	DD	00131	16\$: PUSHL	RET_STATUS	0428		
		66		01	FB	00133	17\$: 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

RPG\$UPDATE
1-002

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

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

Page 11
(4)

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
;\$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 MSRC\$: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

0332 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

