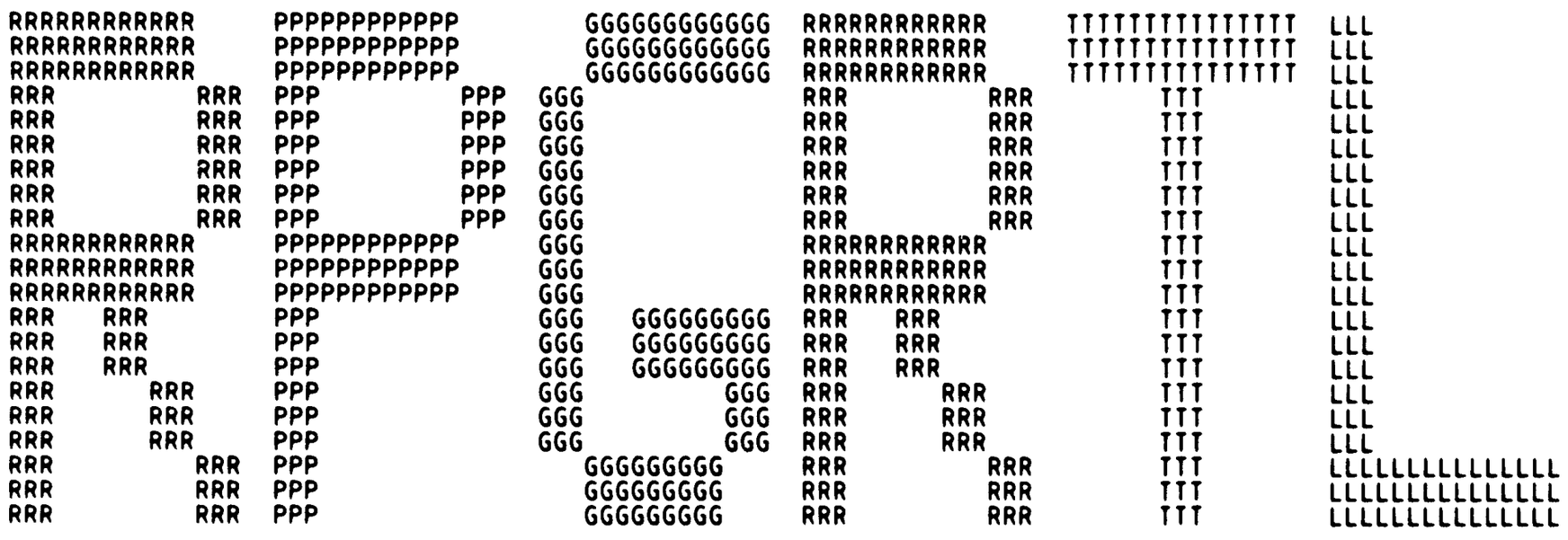
RRRRRRRRRRR PPPPPPPPPPPP GGGGGGGGGGG RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
RRR RRR PPP PPP GGG RRR RRR TTT LLL
RRRRRRRRRRR PPPPPPPPPPP GGG RRRRRRRRRRR TTT LLL
RRR RRR PPP GGG GGGGGGGGG RRR RRR TTT LLL
RRR RRR PPP GGG GGGGGGGGG RRR RRR TTT LLL
RRR RRR PPP GGG GGG RRR RRR TTT LLL
RRR RRR PPP GGG GGG RRR RRR TTT LLL
RRR RRR PPP GGG GGG RRR RRR TTT LLLLLLLLLLLLLLL
RRR RRR PPP GGGGGGGGG RRR RRR TTT LLLLLLLLLLLLLLL
RRR RRR PPP GGGGGGGGG RRR RRR TTT LLLLLLLLLLLLLLL

```
RRRRRRRR    PPPPPPPP      GGGGGGGG  DDDDDDDD      SSSSSSSS  PPPPPPPP    LL          YY      YY
RRRRRRRR    PPPPPPPP      GGGGGGGG  DDDDDDDD      SSSSSSSS  PPPPPPPP    LL          YY      YY
RR      RR  PP      PP  GG          DD      DD  SS          PP      PP  LL          YY      YY
RR      RR  PP      PP  GG          DD      DD  SS          PP      PP  LL          YY      YY
RR      RR  PP      PP  GG          DD      DD  SS          PP      PP  LL            YY  YY
RR      RR  PP      PP  GG          DD      DD  SS          PP      PP  LL            YY  YY
RRRRRRRR    PPPPPPPP    GG          DD      DD    SSSSSS    PPPPPPPP    LL              YY
RRRRRRRR    PPPPPPPP    GG          DD      DD    SSSSSS    PPPPPPPP    LL              YY
RR  RR      PP          GG  GGGGGG  DD      DD          SS  PP          LL              YY
RR  RR      PP          GG  GGGGGG  DD      DD          SS  PP          LL              YY
RR    RR    PP          GG      GG  DD      DD          SS  PP          LL              YY          ....
RR    RR    PP          GG      GG  DD      DD          SS  PP          LL              YY          ....
RR      RR  PP            GGGGGG    DDDDDDDD    SSSSSSSS  PP          LLLLLLLLLL      YY          ....
RR      RR  PP            GGGGGG    DDDDDDDD    SSSSSSSS  PP          LLLLLLLLLL      YY          ....

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
```

```
;     1          0001  0 MODULE RPG$DSPLY( %TITLE 'DSPLY an item'
;     2          0002  0                  IDENT = '1-004'                    ! file RPGDISPLY.B32 EDIT:DG1004
;     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 COPYRIGH) 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 module contains the RTL routine that supports the RPG DSPLY
;    37          0037  1 !       opcode.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:  VAX/VMS user mode
;    40          0040  1 !
;    41          0041  1 ! AUTHOR: Debess Grabazs, CREATION DATE: 18-March-1983
;    42          0042  1 !
;    43          0043  1 ! MODIFIED BY:
;    44          0044  1 !
;    45          0045  1 ! 1-001 - Original version. DG 18-Mar-1983.
;    46          0046  1 ! 1-002 - use routine COB$$RET_A_AB_PREV to get address of COB$$AB_PREV.
;    47          0047  1 !         MDL 29-Aug-1983
;    48          0048  1 ! 1-003 - Create temporary for overpunched numeric before calling
;    49          0049  1 !         COB$ACC_SCR.  In case of invalid entry, original value will
;    50          0050  1 !         be saved. DG 30-Aug-1983.
;    51          0051  1 ! 1-004 - Set RPG for calls to both COB$ACC_SCR and COB$DISP_SCR.
;    52          0052  1 !         DG 24-Oct-1983.
;    53          0053  1 !
;    54          0054  1 !--
;    55          0055  1 !<BLF/PAGE>
```

```
;    57          0056  1 %SBTTL 'Declarations'
;    58          0057  1 !+
;    59          0058  1 ! PROLOGUE FILE:
;    60          0059  1 !-
;    61          0060  1
;    62          0061  1 REQUIRE 'RTLIN:RPGPROLOG';                  ! Switches, PSECTs, macros,
;    63          0126  1                                             !  linkages and LIBRARYs
;    64          0127  1
;    65          0128  1 !+
;    66          0129  1 ! LINKAGES
;    67          0130  1 !       NONE
;    68          0131  1 !-
;    69          0132  1
;    70          0133  1 !+
;    71          0134  1 ! TABLE OF CONTENTS:
;    72          0135  1 !-
;    73          0136  1
;    74          0137  1 FORWARD ROUTINE
;    75          0138  1         RPG$DSPLY: NOVALUE;
;    76          0139  1
;    77          0140  1 !+
;    78          0141  1 ! INCLUDE FILES
;    79          0142  1 !       NONE
;    80          0143  1 !-
;    81          0144  1
;    82          0145  1 !+
;    83          0146  1 ! MACROS
;    84          0147  1 !       NONE
;    85          0148  1 !-
;    86          0149  1
;    87          0150  1 !+
;    88          0151  1 ! EQUATED SYMBOLS
;    89          0152  1 !       NONE
;    90          0153  1 !-
;    91          0154  1
;    92          0155  1 !+
;    93          0156  1 ! EXTERNAL REFERENCES
;    94          0157  1 !-
;    95          0158  1
;    96          0159  1 EXTERNAL ROUTINE
;    97          0160  1         COB$ACC_SCR,                        ! ACCEPT with conversion
;    98          0161  1         COB$DISP_SCR,                       ! DISPLAY with conversion
;    99          0162  1         COB$$RET_A_AB_PREV,                 ! Return address of table giving
;   100          0163  1                                             ! call history for advancing purposes
;   101          0164  1         LIB$SIGNAL;                         ! Signal warning and continue execution
;   102          0165  1
;   103          0166  1 EXTERNAL LITERAL
;   104          0167  1         RPG$_INVNUMENT;                     ! Invalid numeric entry warning
;   105          0168  1
;   106          0169  1 EXTERNAL
;   107          0170  1         RPG$BTZ;                            ! Table for translate blank to zero
```

```
;  109          0171  1 %SBTTL 'RPG$DSPLY - DSPLY an item'
;  110          0172  1 GLOBAL ROUTINE RPG$DSPLY(
;  111          0173  1                         FLAGS,
;  112          0174  1                         DISPLAY_FIELD:  REF BLOCK[,BYTE],
;  113          0175  1                         ACCEPT_FIELD:   REF BLOCK[,BYTE]
;  114          0176  1                         ): NOVALUE =
;  115          0177  1
;  116          0178  1 !++
;  117          0179  1 !
;  118          0180  1 ! FUNCTIONAL DESCRIPTION:
;  119          0181  1 !
;  120          0182  1 !       This routine supports the RPG DSPLY opcode.
;  121          0183  1 !
;  122          0184  1 ! CALLING SEQUENCE:
;  123          0185  1 !
;  124          0186  1 !       CALL RPG$DSPLY (flags.fl.v [,display_field.rx.dx]
;  125          0187  1 !                       [,accept_field.mx.dx])
;  126          0188  1 !
;  127          0189  1 ! FORMAL PARAMETERS:
;  128          0190  1 !
;  129          0191  1 !       flags                   bit 0 is set on if a comma should be
;  130          0192  1 !                               used in place of decimal point; bit 1
;  131          0193  1 !                               is set on if blanks in an overpunched
;  132          0194  1 !                               numeric field should be treated as
;  133          0195  1 !                               equivalent to zeroes.
;  134          0196  1 !
;  135          0197  1 !       display_field           data item whose value is displayed.
;  136          0198  1 !
;  137          0199  1 !       accept_field            data item whose value is displayed and
;  138          0200  1 !                               a new value is accepted into.
;  139          0201  1 !
;  140          0202  1 ! IMPLICIT INPUTS:
;  141          0203  1 !
;  142          0204  1 !       NONE
;  143          0205  1 !
;  144          0206  1 ! IMPLICIT OUTPUTS:
;  145          0207  1 !
;  146          0208  1 !       NONE
;  147          0209  1 !
;  148          0210  1 ! ROUTINE VALUE:
;  149          0211  1 !
;  150          0212  1 !       NONE
;  151          0213  1 !
;  152          0214  1 ! SIDE EFFECTS:
;  153          0215  1 !
;  154          0216  1 !       NONE
;  155          0217  1 !
;  156          0218  1 !--
```

```
;  158     0219  2     BEGIN
;  159     0220  2
;  160     0221  2     LITERAL
;  161     0222  2         TRUE                = 1,
;  162     0223  2         DISP                = 0,
;  163     0224  2
;  164     0225  2         DEC_IS_COMMA_BIT= 1,
;  165     0226  2         BTZ_BIT             = 2,
;  166     0227  2         CONVERT_BIT         = 32,
;  167     0228  2         COMMA_BIT           = 64,
;  168     0229  2         PROTECT_BIT         = 256,
;  169     0230  2         RPG_BIT             = 2048;
;  170     0231  2
;  171     0232  2     LOCAL
;  172     0233  2         COB$$AB_PREV,                       ! Call history for advancing purposes
;  173     0234  2         DEV :   WORD,                       ! 2-byte array -
;  174     0235  2                                             !  device number + error
;  175     0236  2                                             !  handling decider
;  176     0237  2         FLAG,                               ! Attributes bit vector
;  177     0238  2         NRO_FLAG : INITIAL (0),             ! Indicates if dealing with overpunched numeric
;  178     0239  2         TEMP_NRO : BLOCK [12,BYTE] VOLATILE,! Temporary string descriptor
;  179     0240  2         TEMP_STRING : VECTOR [15,BYTE];     ! Temporary string - this assumes an overpunched
;  180     0241  2                                             !  numeric string will only have, at most, 15 digits,
;  181     0242  2                                             !  a sign and a decimal point
;  182     0243  2     BUILTIN
;  183     0244  2         ACTUALCOUNT;
;  184     0245  2
;  185     0246  2     !+
;  186     0247  2     !
;  187     0248  2     ! Get address of history of previous call.
;  188     0249  2     ! Note that using it this way will affect the whole longword
;  189     0250  2     ! at the address, and not just the first byte.
;  190     0251  2     !
;  191     0252  2     !-
;  192     0253  2     COB$$AB_PREV = COB$$RET_A_AB_PREV();
;  193     0254  2
;  194     0255  2     !+
;  195     0256  2     !
;  196     0257  2     ! Set up FLAG parameter.
;  197     0258  2     !
;  198     0259  2     !-
;  199     0260  2     FLAG = CONVERT_BIT + RPG_BIT;               ! Conversion and RPG
;  200     0261  2     IF (.FLAGS AND DEC_IS_COMMA_BIT) NEQ 0
;  201     0262  2     THEN FLAG = .FLAG + COMMA_BIT;              ! Decimal is comma
;  202     0263  2
;  203     0264  2     !+
;  204     0265  2     !
;  205     0266  2     ! Deal with DISPLAY_FIELD.
;  206     0267  2     !
;  207     0268  2     !-
;  208     0269  2     IF .DISPLAY_FIELD NEQ 0
;  209     0270  2     THEN
;  210     0271  3         BEGIN
;  211     0272  3
;  212     0273  3         IF (.DISPLAY_FIELD[DSC$B_DTYPE] EQL DSC$K_DTYPE_NRO) AND
;  213     0274  4            ((.FLAGS AND BTZ_BIT) NEQ 0)
;  214     0275  3         THEN
```

```
;  215     0276  3              !+
;  216     0277  3              ! Convert blanks to zeroes if flag is set.
;  217     0278  3              !-
;  218     0279  3              CH$TRANSLATE (RPG$BTZ,
;  219     0280  3                            .DISPLAY_FIELD[DSC$W_LENGTH], .DISPLAY_FIELD[DSC$A_POINTER],
;  220     0281  3                            0, .DISPLAY_FIELD[DSC$W_LENGTH], .DISPLAY_FIELD[DSC$A_POINTER]);
;  221     0282  3
;  222     0283  3          .COB$$AB_PREV = DISP;                  ! Set up history for proper advancing
;  223     0284  3          !+
;  224     0285  3          ! Display the field.
;  225     0286  3          !-
;  226     0287  3          COB$DISP_SCR (1, .DISPLAY_FIELD, .FLAG);
;  227     0288  3
;  228     0289  2          END;
;  229     0290  2
;  230     0291  2      !+
;  231     0292  2      !
;  232     0293  2      ! Deal with ACCEPT_FIELD.
;  233     0294  2      !
;  234     0295  2      !-
;  235     0296  3      IF  (ACTUALCOUNT() EQL 3 AND .ACCEPT_FIELD NEQ 0)
;  236     0297  2      THEN
;  237     0298  3          BEGIN
;  238     0299  3
;  239     0300  3          IF (.ACCEPT_FIELD[DSC$B_DTYPE] EQL DSC$K_DTYPE_NRO) AND
;  240     0301  4             ((.FLAGS AND BTZ_BIT) NEQ 0)
;  241     0302  3          THEN
;  242     0303  3              !+
;  243     0304  3              ! Convert blanks to zeroes if flag is set.
;  244     0305  3              !-
;  245     0306  3              CH$TRANSLATE (RPG$BTZ,
;  246     0307  3                            .ACCEPT_FIELD[DSC$W_LENGTH], .ACCEPT_FIELD[DSC$A_POINTER],
;  247     0308  3                            0, .ACCEPT_FIELD[DSC$W_LENGTH], .ACCEPT_FIELD[DSC$A_POINTER]);
;  248     0309  3
;  249     0310  3          .COB$$AB_PREV = DISP;                  ! Set up history for proper advancing
;  250     0311  3          !+
;  251     0312  3          ! Display the field.
;  252     0313  3          !-
;  253     0314  3          COB$DISP_SCR (1, .ACCEPT_FIELD, .FLAG);
;  254     0315  3
;  255     0316  3          FLAG = .FLAG OR PROTECT_BIT;           ! Protect
;  256     0317  3          WHILE TRUE DO
;  257     0318  3          !+
;  258     0319  3          ! Keep ACCEPTing until successful.
;  259     0320  3          !-
;  260     0321  4          BEGIN
;  261     0322  4
;  262     0323  4              BIND
;  263     0324  4                  ZERO = UPLIT ('0');
;  264     0325  4
;  265     0326  4              .COB$$AB_PREV = DISP;              ! Set up history for proper advancing
;  266     0327  4              IF .ACCEPT_FIELD [DSC$B_DTYPE] EQL DSC$K_DTYPE_NRO
;  267     0328  4              THEN
;  268     0329  4                  NRO_FLAG = TRUE;
;  269     0330  4              IF .NRO_FLAG
;  270     0331  4              THEN
;  271     0332  5                  BEGIN
```

```
;  272     0333  5                !+
;  273     0334  5                ! Overpunched numeric string must be saved because
;  274     0335  5                ! COB$ACC_SCR zeroes it out and if there is invalid
;  275     0336  5                ! input, the original string must be available in
;  276     0337  5                ! case the user responds to the reprompt with a <CR>.
;  277     0338  5                !-
;  278     0339  5                TEMP_NRO [DSC$W_LENGTH] = .ACCEPT_FIELD [DSC$W_LENGTH];
;  279     0340  5                TEMP_NRO [DSC$B_DTYPE] = DSC$K_DTYPE_NRO;
;  280     0341  5                TEMP_NRO [DSC$A_POINTER] = TEMP_STRING;
;  281     0342  5                IF .ACCEPT_FIELD [DSC$B_CLASS] EQL DSC$K_CLASS_SD
;  282     0343  5                THEN
;  283     0344  6                    BEGIN
;  284     0345  6
;  285     0346  6                    TEMP_NRO [DSC$B_CLASS] = DSC$K_CLASS_SD;
;  286     0347  6                    TEMP_NRO [DSC$B_SCALE] = .ACCEPT_FIELD [DSC$B_SCALE];
;  287     0348  6                    TEMP_NRO [DSC$B_DIGITS] = .ACCEPT_FIELD [DSC$B_DIGITS];
;  288     0349  6
;  289     0350  6                    END
;  290     0351  5                ELSE
;  291     0352  5                    TEMP_NRO [DSC$B_CLASS] = DSC$K_CLASS_S;
;  292     0353  5
;  293     0354  5                !+
;  294     0355  5                ! Initialize the temporary string.
;  295     0356  5                !-
;  296     0357  5                CH$MOVE (.ACCEPT_FIELD [DSC$W_LENGTH],
;  297     0358  5                         .ACCEPT_FIELD [DSC$A_POINTER],
;  298     0359  5                         TEMP_STRING);
;  299     0360  5
;  300     0361  4                END;
;  301     0362  4
;  302     0363  4            !+
;  303     0364  4            ! Accept into the field.
;  304     0365  4            !-
;  305     0366  6                IF ( COB$ACC_SCR (0, (IF .NRO_FLAG
;  306     0367  6                                      THEN TEMP_NRO
;  307     0368  5                                      ELSE .ACCEPT_FIELD),
;  308     0369  5                                 .FLAG, 0, 0, 0, 0) )
;  309     0370  4                THEN EXITLOOP
;  310     0371  4                ELSE
;  311     0372  5                    BEGIN
;  312     0373  5                    !+
;  313     0374  5                    ! Error - print warning and re-prompt.
;  314     0375  5                    !-
;  315     0376  5                    LIB$SIGNAL (RPG$_INVNUMENT);
;  316     0377  5                    .COB$$AB_PREV = DISP;           ! Set up history for proper advancing
;  317     0378  5
;  318     0379  4                    END;
;  319     0380  4
;  320     0381  3            END;
;  321     0382  3
;  322     0383  3            IF .NRO_FLAG
;  323     0384  3            THEN
;  324     0385  3                !+
;  325     0386  3                ! Move accepted string from temporary to actual accept field.
;  326     0387  3                !-
;  327     0388  3                CH$MOVE (.ACCEPT_FIELD [DSC$W_LENGTH], TEMP_STRING,
;  328     0389  3                                .ACCEPT_FIELD [DSC$A_POINTER]);
```

```
;  329          0390  3
;  330          0391  2          END;
;  331          0392  2
;  332          0393  1      END;


                                                        .TITLE  RPG$DSPLY DSPLY an item
                                                        .IDENT  \1-004\

                                                        .PSECT  _RPG$CODE,NOWRT,  SHR,  PIC,2

                                    00  00  00  30  00000 P.AAA:  .ASCII  \0\<0><0><0>                                ;
                                                        ZERO=           P.AAA
                                                        .EXTRN  COB$ACC_SCR, COB$DISP_SCR
                                                        .EXTRN  COB$$RET_A_AB_PREV
                                                        .EXTRN  LIB$SIGNAL, RPG$_INVNUMENT
                                                        .EXTRN  RPG$BTZ

                                                        .ENTRY  RPG$DSPLY, Save R2,R3,R4,R5,R6,R7,R8,R9,-       ; 0172
                                                                R10,R11
                                    5B 00000000G  00  9E 00002          MOVAB   COB$DISP_SCR, R11                   ;
                                    5A 00000000G  00  9E 00009          MOVAB   RPG$BTZ, R10                        ;
                                    5E            1C  C2 00010          SUBL2   #28, SP                             ;
                                                  58  D4 00013          CLRL    NRO_FLAG                            ; 0219
                                00000000G   00    00  FB 00015          CALLS   #0, COB$$RET_A_AB_PREV              ; 0253
                                    59            50  D0 0001C          MOVL    R0, COB$$AB_PREV                    ;
                                    57      0820  8F  3C 0001F          MOVZWL  #2080, FLAG                         ; 0260
                                    04        04  AC  E9 00024          BLBC    FLAGS, 1$                           ; 0261
                                    57        40  A7  9E 00028          MOVAB   64(R7), FLAG                        ; 0262
                                    56        08  AC  D0 0002C 1$:      MOVL    DISPLAY_FIELD, R6                   ; 0269
                                                  1D  13 00030          BEQL    3$                                  ;
                                    13        02  A6  91 00032          CMPB    2(R6), #19                          ; 0273
                                                  0D  12 00036          BNEQ    2$                                  ;
                                          09      6C  21  E1 00038          BBC     #33, FLAGS, 2$                  ; 0274
                                6A        00  04  B6  66  2E 0003C          MOVTC   (R6), @4(R6), #0, RPG$BTZ, (R6), @4(R6) ; 0281
                                              04  B6  66     00042
                                                  69  D4 00045 2$:      CLRL    (COB$$AB_PREV)                      ; 0283
                                    7E            56  7D 00047          MOVQ    R6, -(SP)                           ; 0287
                                                  01  DD 0004A          PUSHL   #1                                  ;
                                    6B            03  FB 0004C          CALLS   #3, COB$DISP_SCR                    ;
                                    03            6C  91 0004F 3$:      CMPB    (AP), #3                            ; 0296
                                                  01  13 00052          BEQL    4$                                  ;
                                                      04 00054          RET                                         ;
                                          0C      AC  D5 00055 4$:      TSTL    ACCEPT_FIELD                        ;
                                                  01  12 00058          BNEQ    5$                                  ;
                                                      04 0005A          RET                                         ;
                                    56    0C      AC  D0 0005B 5$:      MOVL    ACCEPT_FIELD, R6                    ; 0300
                                    13    02      A6  91 0005F          CMPB    2(R6), #19                          ;
                                                  0D  12 00063          BNEQ    6$                                  ;
                                          09      6C  21  E1 00065          BBC     #33, FLAGS, 6$                  ; 0301
                                6A        00  04  B6  66  2E 00069          MOVTC   (R6), @4(R6), #0, RPG$BTZ, (R6), @4(R6) ; 0308
                                              04  B6  66     0006F
                                                  69  D4 00072 6$:      CLRL    (COB$$AB_PREV)                      ; 0310
                                    7E            56  7D 00074          MOVQ    R6, -(SP)                           ; 0314
                                                  01  DD 00077          PUSHL   #1                                  ;
                                    6B            03  FB 00079          CALLS   #3, COB$DISP_SCR                    ;
```

```
                                57      0100    8F  A8 0007C            BISW2   #256, FLAG                              ; 0316
                                                69  D4 00081 7$:        CLRL    (COB$$AB_PREV)                          ; 0326
                                13        02    A6  91 00083            CMPB    2(R6), #19                              ; 0327
                                                03  12 00087            BNEQ    8$                                      ;
                                58              01  D0 00089            MOVL    #1, NRO_FLAG                            ; 0329
                                2B              58  E9 0008C 8$:        BLBC    NRO_FLAG, 11$                           ; 0330
                            10  AE              66  B0 0008F            MOVW    (R6), TEMP_NRO                          ; 0339
                            12  AE              13  90 00093            MOVB    #19, TEMP_NRO+2                         ; 0340
                            14  AE              6E  9E 00097            MOVAB   TEMP_STRING, TEMP_NRO+4                 ; 0341
                                09        03    A6  91 0009B            CMPB    3(R6), #9                               ; 0342
                                                10  12 0009F            BNEQ    9$                                      ;
                            13  AE              09  90 000A1            MOVB    #9, TEMP_NRO+3                          ; 0346
                            18  AE        08    A6  90 000A5            MOVB    8(R6), TEMP_NRO+8                       ; 0347
                            19  AE        09    A6  90 000AA            MOVB    9(R6), TEMP_NRO+9                       ; 0348
                                                04  11 000AF            BRB     10$                                     ; 0342
                            13  AE              01  90 000B1 9$:        MOVB    #1, TEMP_NRO+3                          ; 0352
                    6E      04  B6              66  28 000B5 10$:       MOVC3   (R6), @4(R6), TEMP_STRING               ; 0357
                                                7E  7C 000BA 11$:       CLRQ    -(SP)                                   ; 0366
                                                7E  7C 000BC            CLRQ    -(SP)                                   ;
                                                57  DD 000BE            PUSHL   FLAG                                    ; 0369
                                08              58  E9 000C0            BLBC    NRO_FLAG, 12$                           ;
                                50        24    AE  9E 000C3            MOVAB   TEMP_NRO, R0                            ; 0366
                                                50  DD 000C7            PUSHL   R0                                      ;
                                                02  11 000C9            BRB     13$                                     ;
                                                56  DD 000CB 12$:       PUSHL   R6                                      ; 0368
                                                7E  D4 000CD 13$:       CLRL    -(SP)                                   ; 0366
                    00000000G   00              07  FB 000CF            CALLS   #7, COB$ACC_SCR                         ;
                                11              50  E8 000D6            BLBS    R0, 14$                                 ;
                                    00000000G   8F  DD 000D9            PUSHL   #RPG$_INVNUMENT                         ; 0376
                    00000000G   00              01  FB 000DF            CALLS   #1, LIB$SIGNAL                          ;
                                                69  D4 000E6            CLRL    (COB$$AB_PREV)                          ; 0377
                                                97  11 000E8            BRB     7$                                      ; 0317
                                05              58  E9 000EA 14$:       BLBC    NRO_FLAG, 15$                           ; 0383
                04  B6              6E          66  28 000ED            MOVC3   (R6), TEMP_STRING, @4(R6)               ; 0389
                                                    04 000F2 15$:       RET                                             ; 0393

; Routine Size:  243 bytes,    Routine Base:  _RPG$CODE + 0004

;  333          0394  1
;  334          0395  0 END ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _RPG$CODE                     247  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                               Library Statistics
```

```
;
;
;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1   9776         9         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$:RPGDSPLY/OBJ=OBJ$:RPGDSPLY MSRC$:RPGDSPLY/UPDATE=(ENH$:RPGDSPLY)
;
; Size:          243 code + 4 data bytes
; Run Time:        00:07.3
; Elapsed Time:    00:25.5
; Lines/CPU Min:    3259
; Lexemes/CPU-Min: 14079
; Memory Used:  103 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RMS0TRUNC
LIS

STAPRFLNM
LIS

RPGCVTPTO
LIS

RPGHANDLE
LIS

RPGMOVE1
LIS

RPGLIB
REQ

RMSGBL
LIS

RPGRTL

RPGDSPLY
LIS

RPGRTL
MAP

RPGPROLOG
REQ

RPGEXTIND
LIS

RPGLIB
LIS

RMS0SRCH
LIS

RMS0WAIT
LIS

RPGBTZ
LIS

RPGMOVE2
LIS

RPGDIVIDE
LIS

RPGIOEXCE
LIS

RPGDEF
REQ

RMS0UPDAT
LIS

RPGERROR
LIS