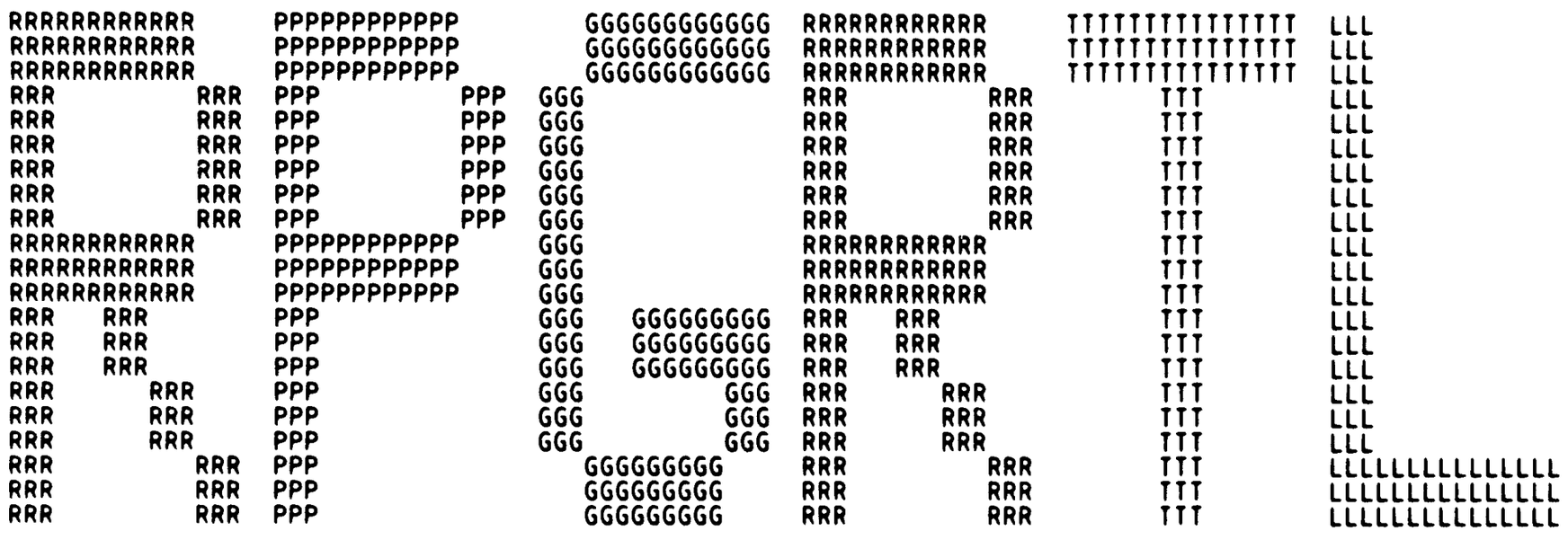
RRRRRRRRRRR PPPPPPPPPPPP GGGGGGGGGGG RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
RRR RRR PPP PPP GGG RRR RRR TTT LLL
RRRRRRRRRRR PPPPPPPPPPP GGG RRRRRRRRRRR TTT LLL
RRR RRR PPP GGG GGGGGGGGG RRR RRR TTT LLL
RRR RRR PPP GGG GGGGGGGGG RRR RRR TTT LLL
RRR RRR PPP GGG GGG RRR RRR TTT LLL
RRR RRR PPP GGG GGG RRR RRR 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  HH      HH    AAAAAA    NN      NN  DDDDDDDD    LL          EEEEEEEEEE
RRRRRRRR    PPPPPPPP      GGGGGGGG  HH      HH    AAAAAA    NN      NN  DDDDDDDD    LL          EEEEEEEEEE
RR      RR  PP      PP  GG          HH      HH  AA      AA  NN      NN  DD      DD  LL          EE
RR      RR  PP      PP  GG          HH      HH  AA      AA  NNNN    NN  DD      DD  LL          EE
RR      RR  PP      PP  GG          HH      HH  AA      AA  NNNN    NN  DD      DD  LL          EE
RRRRRRRR    PPPPPPPP    GG          HHHHHHHHHH  AA      AA  NN  NN  NN  DD      DD  LL          EEEEEEEE
RRRRRRRR    PPPPPPPP    GG          HHHHHHHHHH  AA      AA  NN  NN  NN  DD      DD  LL          EEEEEEEE
RR  RR      PP          GG  GGGGGG  HH      HH  AAAAAAAAAA  NN    NNNN  DD      DD  LL          EE
RR  RR      PP          GG  GGGGGG  HH      HH  AAAAAAAAAA  NN    NNNN  DD      DD  LL          EE
RR    RR    PP          GG      GG  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE
RR    RR    PP          GG      GG  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE
RR      RR  PP            GGGGGG    HH      HH  AA      AA  NN      NN  DDDDDDDD    LLLLLLLLLL  EEEEEEEEEE
RR      RR  PP            GGGGGG    HH      HH  AA      AA  NN      NN  DDDDDDDD    LLLLLLLLLL  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
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE RPG$HANDLER ( %TITLE 'RPG error handler'
;     2          0002  0                 IDENT = '1-003'         ! FILE: RPGHANDLE.B32 EDIT:DG1003
;     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 ! FACILITY:     RPGII SUPPORT
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:     This procedure is the error handler for RPGII error
;    34          0034  1 !               conditions.  It gets invoked as a result of a call
;    35          0035  1 !               to LIB$SIGNAL.
;    36          0036  1 !
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  Vax-11 User Mode
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Debess Grabazs, CREATION DATE: 18-JAN-1983
;    41          0041  1 !
;    42          0042  1   MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original. DG 18-JAN-1983
;    45          0045  1 ! 1-002 - Added conversion of COB$ error codes to RPG$ equivalents.
;    46          0046  1 !         DG 31-MAY-1983
;    47          0047  1 ! 1-003 - Make sure conversion of error codes generated by COBOL to RPG$
;    48          0048  1 !         equivalents is only done when called by RPG$DSPLY. DG 12-OCT-1983
;    49          0049  1 !
;    50          0050  1 !--
;    51          0051  1
;    52          0052  1 !<BLF/PAGE>
```

```
;    54          0053  1 %SBTTL 'Declarations'
;    55          0054  1 !+
;    56          0055  1 ! PROLOGUE FILE:
;    57          0056  1 !-
;    58          0057  1
;    59          0058  1 REQUIRE 'RTLIN:RPGPROLOG';                  ! Switches, PSECTs, macros,
;    60          0123  1                                            !  linkages and LIBRARYs
;    61          0124  1
;    62          0125  1 !+
;    63          0126  1 ! LINKAGES
;    64          0127  1 !       NONE
;    65          0128  1 !-
;    66          0129  1
;    67          0130  1 !+
;    68          0131  1 ! TABLE OF CONTENTS:
;    69          0132  1 !-
;    70          0133  1
;    71          0134  1 FORWARD ROUTINE
;    72          0135  1         RPG$HANDLER ;
;    73          0136  1
;    74          0137  1 !+
;    75          0138  1 ! INCLUDE FILES
;    76          0139  1 !       NONE
;    77          0140  1 !-
;    78          0141  1
;    79          0142  1 !+
;    80          0143  1 ! MACROS
;    81          0144  1 !       NONE
;    82          0145  1 !-
;    83          0146  1
;    84          0147  1 !+
;    85          0148  1 ! EQUATED SYMBOLS
;    86          0149  1 !-
;    87          0150  1
;    88          0151  1 LITERAL
;    89          0152  1         CVTTP_OPCODE = %X'26',          ! Opcode value for CVTTP instruction
;    90          0153  1         CVTSP_OPCODE = %X'09';          ! Opcode value for CVTSP instruction
;    91          0154  1
;    92          0155  1 !+
;    93          0156  1 ! EXTERNAL REFERENCES
;    94          0157  1 !-
;    95          0158  1
;    96          0159  1 EXTERNAL ROUTINE
;    97          0160  1         LIB$SIGNAL ;                    ! Stop execution via signalling
;    98          0161  1
;    99          0162  1 EXTERNAL LITERAL
;   100          0163  1         COB$_EOFON_ACC ,                ! EOF on ACCEPT
;   101          0164  1         COB$_ERRDURACC ,                ! Error during ACCEPT
;   102          0165  1         COB$_ERRDURDIS ,                ! Error during DISPLAY
;   103          0166  1         COB$_INVARG ,                   ! Invalid argument
;   104          0167  1         RPG$_ERRDURDSP ,                ! Error during DSPLY signal
;   105          0168  1         RPG$_INVARG ,                   ! Invalid argument signal
;   106          0169  1         RPG$_INVARRIND ,                ! Invalid array index signal
;   107          0170  1         RPG$_INVNUMFLD ;                ! Invalid numeric field signal
;   108          0171  1
```

```
;  110          0172  1 %SBTTL 'RPG$HANDLER - RPG error handler'
;  111          0173  1 GLOBAL ROUTINE RPG$HANDLER(
;  112          0174  1                           SIGNAL:         REF BLOCK [,BYTE],      ! Nature of the condition
;  113          0175  1                           MECHANISM:      REF BLOCK [,BYTE]       ! State of process at time of signal
;  114          0176  1                           )=
;  115          0177  1
;  116          0178  1 !++
;  117          0179  1 !
;  118          0180  1 ! FUNCTIONAL DESCRIPTION:
;  119          0181  1 !
;  120          0182  1 !       This routine is the error handler for RPGII error conditions.
;  121          0183  1 !       It gets invoked as a result of a call to LIB$SIGNAL.
;  122          0184  1 !
;  123          0185  1 ! CALLING SEQUENCE:
;  124          0186  1 !
;  125          0187  1 !       RPG$HANDLER (signal.rr.r, mechanism.rr.r)
;  126          0188  1 !
;  127          0189  1 ! FORMAL PARAMETERS:
;  128          0190  1 !
;  129          0191  1 !       signal          Address of vector of longwords indicating
;  130          0192  1 !                       the nature of condition.
;  131          0193  1 !
;  132          0194  1 !       mechanism       Address of vector of longwords indicating
;  133          0195  1 !                       the state of the process.
;  134          0196  1 !
;  135          0197  1 ! IMPLICIT INPUTS:
;  136          0198  1 !
;  137          0199  1 !       NONE
;  138          0200  1 !
;  139          0201  1 ! IMPLICIT OUTPUTS:                                                                              ;
;  140          0202  1 !
;  141          0203  1 !       NONE
;  142          0204  1 !                                                                                               ;
;  143          0205  1 ! ROUTINE VALUE:                                                                                ;
;  144          0206  1 !
;  145          0207  1 !       SS$_RESIGNAL
;  146          0208  1 !
;  147          0209  1 ! COMPLETION CODES:
;  148          0210  1 !
;  149          0211  1 !       NONE
;  150          0212  1 !
;  151          0213  1 ! SIDE EFFECTS:                                                                                 ;
;   52          0214  1 !                                                                                               ;
;  153          0215  1 !       NONE                                                                                    ;
;  154          0216  1 !                                                                                               ;
;  155          0217  1 ! NOTES:                                                                                        ;
;  156          0218  1 !
;  157          0219  1 !       The macro field references beginning with 'CHF' refer to
;  158          0220  1 !       condition handling argument list offsets.
;  159          0221  1 !                                                                                               ;
;  160          0222  1 !--                                                                                             ;
```

```
;  162     0223  2     BEGIN
;  163     0224  2
;  164     0225  2     LOCAL
;  165     0226  2         CONDITION,                      ! Condition that was signalled
;  166     0227  2         USER_PC;                        ! Program counter where exception took
;  167     0228  2                                         ! place
;  168     0229  2
;  169     0230  2     !+
;  170     0231  2     ! Find out if it is a signal of interest.
;  171     0232  2     !-
;  172     0233  2
;  173     0234  2     CONDITION = .SIGNAL [CHF$L_SIG_NAME];       ! Fetch condition value from signal array
;  174     0235  2
;  175     0236  2     !+
;  176     0237  2     ! Select appropriate action based on which one we have.
;  177     0238  2     !-
;  178     0239  2
;  179     0240  2     SELECTONE .CONDITION OF
;  180     0241  2       SET
;  181     0242  2
;  182     0243  2         [SS$_ROPRAND]:                  ! Was a SS$_ROPRAND
;  183     0244  3             BEGIN
;  184     0245  3
;  185     0246  3                 !+
;  186     0247  3                 ! Using the signal argument vector, extract
;  187     0248  3                 ! the program counter at the time the
;  188     0249  3                 ! SS$_ROPRAND occurred.
;  189     0250  3                 !-
;  190     0251  3
;  191     0252  3                 USER_PC = .SIGNAL [(.SIGNAL [CHF$L_SIG_ARGS] - 1 ) * %UPVAL, 0, %BPVAL, 0] ;
;  192     0253  3
;  193     0254  3                 !+
;  194     0255  3                 ! Check to see if a CVTTP or a CVTSP instruction
;  195     0256  3                 ! was the generator of the signal.  Note that the
;  196     0257  3                 ! PC is pointing to the instruction that caused the
;  197     0258  3                 ! fault.
;  198     0259  3                 !-
;  199     0260  3
;  200     0261  4                 IF (  .(.USER_PC)<0,8> EQL CVTTP_OPCODE  OR
;  201     0262  4                       .(.USER_PC)<0,8> EQL CVTSP_OPCODE     )
;  202     0263  4                 THEN
;  203     0264  4
;  204     0265  4                     !+
;  205     0266  4                     ! Set the first longword of the signal argument
;  206     0267  4                     ! vector (the condition value field) to the
;  207     0268  4                     ! condition being signalled.  In this case, the
;  208     0269  4                     ! condition is invalid decimal data.
;  209     0270  4                     !-
;  210     0271  4
;  211     0272  4                     SIGNAL [CHF$L_SIG_NAME] = RPG$_INVNUMFLD ;
;  212     0273  4
;  213     0274  3
;  214     0275  2             END ;
;  215     0276  2
;  216     0277  2         [SS$_SUBRNG]:           ! Was a SS$_SUBRNG
;  217     0278  3           BEGIN
;  218     0279  3
```

```
;   219     0280  3                 !+
;   220     0281  3                 ! First make sure that the error occurred during a
;   221     0282  3                 ! call from RPG$DSPLY and not from a user subroutine
;   222     0283  3                 !-
;   223     0284  3
;   224     0285  3                 IF .MECHANISM [CHF$L_MCH_DEPTH] EQL 0
;   225     0286  3                 THEN
;   226     0287  3
;   227     0288  3                     !+
;   228     0289  3                     ! Set the first longword of the signal argument
;   229     0290  3                     ! vector (the condition value field) to the
;   230     0291  3                     ! condition being signalled.  In this case, the
;   231     0292  3                     ! condition is subscript out of range.
;   232     0293  3                     !-
;   233     0294  3
;   234     0295  3                     SIGNAL [CHF$L_SIG_NAME] = RPG$_INVARRIND;   ! Set to RPG signal
;   235     0296  2               END;
;   236     0297  2
;   237     0298  2             [COB$_ERRDURACC,
;   238     0299  2              COB$_EOFON_ACC,
;   239     0300  2              COB$_ERRDURDIS]:            ! Was an error during ACCEPT or DISPLAY
;   240     0301  3               BEGIN
;   241     0302  3
;   242     0303  3                     !+
;   243     0304  3                     ! Set the first longword of the signal argument
;   244     0305  3                     ! vector (the condition value field) to the
;   245     0306  3                     ! condition being signalled.
;   246     0307  3                     !-
;   247     0308  3
;   248     0309  3                     SIGNAL [CHF$L_SIG_NAME] = RPG$_ERRDURDSP;       ! Set to RPG signal
;   249     0310  2               END;
;   250     0311  2
;   251     0312  2             [COB$_INVARG]:              ! Was a COB$_INVARG
;   252     0313  3               BEGIN
;   253     0314  3
;   254     0315  3                     !+
;   255     0316  3                     ! Set the first longword of the signal argument
;   256     0317  3                     ! vector (the condition value field) to the
;   257     0318  3                     ! condition being signalled.
;   258     0319  3                     !-
;   259     0320  3
;   260     0321  3                     SIGNAL [CHF$L_SIG_NAME] = RPG$_INVARG;! Set to RPG signal
;   261     0322  2               END;
;   262     0323  2
;   263     0324  2             [OTHERWISE]:                ! No match occurred
;   264     0325  2               0;                        ! No change
;   265     0326  2
;   266     0327  2            TES;
;   267     0328  2
;   268     0329  2         !+
;   269     0330  2         ! Resignal the error with the signal name changed where
;   270     0331  2         ! appropriate.
;   271     0332  2         !-
;   272     0333  2
;   273     0334  2         RETURN SS$_RESIGNAL
;   274     0335  2
;   275     0336  1         END;
```

```
                                                        .TITLE  RPG$HANDLER RPG error handler
                                                        .IDENT  \1-003\

                                                        .EXTRN  LIB$SIGNAL, COB$_EOFON_ACC
                                                        .EXTRN  COB$_ERRDURACC, COB$_ERRDURDIS
                                                        .EXTRN  COB$_INVARG, RPG$_ERRDURDSP
                                                        .EXTRN  RPG$_INVARG, RPG$_INVARRIND
                                                        .EXTRN  RPG$_INVNUMFLD

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

                                                        .ENTRY  RPG$HANDLER, Save R2,R3                 ; 0173
                                52        04    AC  D0 00002        MOVL    SIGNAL, R2                  ; 0234
                                53        04    A2  9E 00006        MOVAB   4(R2), R3
                                51              63  D0 0000A        MOVL    (R3), CONDITION
                    00000454    8F              51  D1 0000D        CMPL    CONDITION, #1108            ; 0243
                                                1B  12 00014        BNEQ    2$
                                50              62  D0 00016        MOVL    (R2), R0                    ; 0252
                                51        FC A240   D0 00019        MOVL    -4(R2)[R0], USER_PC
                                26              61  91 0001E        CMPB    (USER_PC), #38              ; 0261
                                                05  13 00021        BEQL    1$
                                09              61  91 00023        CMPB    (USER_PC), #9               ; 0262
                                                58  12 00026        BNEQ    6$
                                63 00000000G    8F  D0 00028 1$:    MOVL    #RPG$_INVNUMFLD, (R3)       ; 0272
                                                4F  11 0002F        BRB     6$                          ; 0240
                    000004AC    8F              51  D1 00031 2$:    CMPL    CONDITION, #1196            ; 0277
                                                12  12 00038        BNEQ    3$
                                52        08    AC  D0 0003A        MOVL    MECHANISM, R2               ; 0285
                                          08    A2  D5 0003E        TSTL    8(R2)
                                                3D  12 00041        BNEQ    6$
                                63 00000000G    8F  D0 00043        MOVL    #RPG$_INVARRIND, (R3)       ; 0295
                                                34  11 0004A        BRB     6$                          ; 0240
                    00000000G   8F              51  D1 0004C 3$:    CMPL    CONDITION, #COB$_ERRDURACC  ; 0298
                                                12  13 00053        BEQL    4$
                    00000000G   8F              51  D1 00055        CMPL    CONDITION, #COB$_EOFON_ACC
                                                09  13 0005C        BEQL    4$
                    00000000G   8F              51  D1 0005E        CMPL    CONDITION, #COB$_ERRDURDIS
                                                09  12 00065        BNEQ    5$
                                63 00000000G    8F  D0 00067 4$:    MOVL    #RPG$_ERRDURDSP, (R3)       ; 0309
                                                10  11 0006E        BRB     6$                          ; 0240
                    00000000G   8F              51  D1 00070 5$:    CMPL    CONDITION, #COB$_INVARG     ; 0312
                                                07  12 00077        BNEQ    6$
                                63 00000000G    8F  D0 00079        MOVL    #RPG$_INVARG, (R3)          ; 0321
                                50      0918    8F  3C 00080 6$:    MOVZWL  #2328, R0                   ; 0334
                                                    04 00085        RET                                 ; 0336

; Routine Size:  134 bytes,    Routine Base:  _RPG$CODE + 0000

;  276          0337  1
;  277          0338  1 END
;  278          0339  0 ELUDOM
```

; PSECT SUMMARY

Name	Bytes	Attributes
_RPG$CODE	134	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

; Library Statistics

File	Symbols Total	Symbols Loaded	Symbols Percent	Pages Mapped	Processing Time
_255DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:01.)
_255DUA28:[RPGRTL.OBJ]RPGLIB.L32;1	54	0	0	9	00:00.1

; COMMAND QUALIFIERS

```
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:RPGHANDLE/OBJ=OBJ$:RPGHANDLE MSRC$:RPGHANDLE/UPDATE=(ENH$:RPGHANDLE
;       )

; Size:           134 code + 0 data bytes
; Run Time:          00:04.7
; Elapsed Time:      00:18.1
; Lines/CPU Min:      4336
; Lexemes/CPU-Min:    8891
; Memory Used:    51 pages
; Compilation Complete
```

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