I 4

```
PPPPPPPPPPPP     LLL                  IIIIIIIII     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP     LLL                  IIIIIIIII     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP     LLL                  IIIIIIIII     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPP         PPP  LLL                     III        RRR         RRR        TTT        LLL
PPP         PPP  LLL                     III        RRR         RRR        TTT        LLL
PPP         PPP  LLL                     III        RRR         RRR        TTT        LLL
PPP         PPP  LLL                     III        RRR         RRR        TTT        LLL
PPP         PPP  LLL                     III        RRR         RRR        TTT        LLL
PPP         PPP  LLL                     III        RRR         RRR        TTT        LLL
PPPPPPPPPPPP     LLL                     III        RRRRRRRRRRRR           TTT        LLL
PPPPPPPPPPPP     LLL                     III        RRRRRRRRRRRR           TTT        LLL
PPPPPPPPPPPP     LLL                     III        RRRRRRRRRRRR           TTT        LLL
PPP              LLL                     III        RRR   RRR              TTT        LLL
PPP              LLL                     III        RRR   RRR              TTT        LLL
PPP              LLL                     III        RRR   RRR              TTT        LLL
PPP              LLL                     III        RRR      RRR           TTT        LLL
PPP              LLL                     III        RRR      RRR           TTT        LLL
PPP              LLL                     III        RRR      RRR           TTT        LLL
PPP              LLLLLLLLLLLLLLL      IIIIIIIII     RRR         RRR        TTT        LLLLLLLLLLLLLLL
PPP              LLLLLLLLLLLLLLL      IIIIIIIII     RRR         RRR        TTT        LLLLLLLLLLLLLLL
PPP              LLLLLLLLLLLLLLL      IIIIIIIII     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
_$2

Sym
---
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI

PLI
PLI
PLI

PLI
PLI
PLI
PLI
PLI
PLI
PLI

PLI
PLI
PLI
PLI

PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
```

```
**FILE**ID**PLIGETEDI                                    E  4
```

PL
1-

```
PPPPPPPP    LL            IIIIII      GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  EEEEEEEEEE  DDDDDDDD      IIIIII
PPPPPPPP    LL            IIIIII      GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  EEEEEEEEEE  DDDDDDDD      IIIIII
PP      PP  LL              II      GG          EE              TT      EE          DD      DD      II
PP      PP  LL              II      GG          EE              TT      EE          DD      DD      II
PP      PP  LL              II      GG          EE              TT      EE          DD      DD      II
PPPPPPPP    LL              II      GG          EEEEEEEE        TT      EEEEEEEE    DD      DD      II
PPPPPPPP    LL              II      GG          EEEEEEEE        TT      EEEEEEEE    DD      DD      II
PP          LL              II      GG  GGGGGG  EE              TT      EE          DD      DD      II
PP          LL              II      GG  GGGGGG  EE              TT      EE          DD      DD      II
PP          LL              II      GG      GG  EE              TT      EE          DD      DD      II          ....
PP          LL              II      GG      GG  EE              TT      EE          DD      DD      II          ....
PP          LLLLLLLLLL    IIIIII      GGGGGG    EEEEEEEEEE      TT      EEEEEEEEEE  DDDDDDDD      IIIIII        ....
PP          LLLLLLLLLL    IIIIII      GGGGGG    EEEEEEEEEE      TT      EEEEEEEEEE  DDDDDDDD      IIIIII        ....

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
```

```
0000            .title  pli$geteitem
0000            .ident  /1-002/                                 ; Edit WHM1002
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ;
0000
0000    ;++
0000    ; facility:
0000    ;
0000    ;       VAX/VMS PL1 runtime library
0000    ;
0000    ; abstract:
0000    ;
0000    ;       This module contains the pl1 runtime routines to get items from
0000    ;       pl1 stream file under edit controlled format.
0000    ;
0000    ;
0000    ; author: c. spitz 28-nov-79
0000    ;
0000    ; modified:
0000    ;
0000    ;
0000    ;       1-002   Bill Matthews   29-September-1982
0000    ;
0000    ;               Invoke macros $defdat and rtshare instead of $defopr and share.
0000    ;
0000    ;--
0000
0000    ;
0000    ; external definitions
0000    ;
0000            $deffcb                         ;define file control block
0000            $defstk                         ;define stack frame offsets
0000            $defstr                         ;define stream block offsets
```

```
                                 0000          $defdat                     ;define operand node data types
                                 0000          $defgetopt                  ;define get options block
                                 0000          $defcvtind                  ;define convert indices
                                 0000          $rabdef                     ;define rms rab offsets
                                 0000          $rmsdef                     ;define rms error codes
                                 0000
                                 0000    ;
                                 0000    ; local data
                                 0000    ;
                                 0000
                                 0000          rtshare             ;sharable
                                 0000
                                 0000
                                 0000    ;++
                                 0000    ; pli$gete****
                                 0000    ;
                                 0000    ; the pli$gete**** routines are called by the compiled code to get items
                                 0000    ; from a stream input file under edit directed transmission. each routine
                                 0000    ; saves the target item descriptor, calls pli$$getfmt_r6 to get the
                                 0000    ; next field based on the format, and then calls to pli$rt_cvrt_r3 to store
                                 0000    ; the item in the target.
                                 0000    ;--
                                 0000
                                 0000    ;pli$getechar_r6
                                 0000    ; inputs:
                                 0000    ;      r0 - address of element to get
                                 0000    ;      r1 - size/prec of element to get
                                 0000    ;      r11 - address of stream block
                                 0000    ;      ap - address of file control block
                                 0000    ; outputs:
                                 0000    ;      none
                                 0000    ; side effects:
                                 0000    ;      r0-r6 are destroyed
                                 0000
                                 0000    pli$getechar_r6::
           0C AC    08     C8    0000          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
   0C AC   04000000 8F     CA    0004          bicl    #atr_m_flttrg,fcb_l_attr(ap) ;clr floating target flag
              7E    50     7D    000C          movq    r0,-(sp)                    ;save destination
       00000000'GF  16     16    000F          jsb     g^pli$$getfmt_r6            ;get format
              52    8E     7D    0015          movq    (sp)+,r2                    ;restore destination
              54    05     C0    0018          addl    #cvt_k_dst_char,r4          ;update case index for char dest
     00000000'GF    00     FB    001B          calls   #0,g^pli$cvrt_cg_r3         ;convert it
           0C AC    08     CA    0022          bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                           05    0026          rsb                                 ;return
                                 0027
                                 0027    ;pli$getevcha_r6
                                 0027    ;   inputs:
                                 0027    ;      r0 - address of element to get
                                 0027    ;      r1 - size/prec of element to get
                                 0027    ;      r11 - address of stream block
                                 0027    ;      ap - address of file control block
                                 0027    ;   outputs:
                                 0027    ;      none
                                 0027    ;   side effects:
                                 0027    ;      r0-r6 are destroyed
                                 0027    ;
                                 0027
```

```
                              0027  pli$getevcha_r6::
         0C AC      08     C8 0027          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
   0C AC    04000000 8F    CA 002B          bicl    #atr_m_flttrg,fcb_l_attr(ap) ;clr floating target flag
                    60     B4 0033          clrw    (r0)                        ;clear length
              7E    50     7D 0035          movq    r0,-(sp)                    ;save destination
         00000000'GF       16 0038          jsb     g^pli$$getfmt_r6            ;get format
              52    8E     7D 003E          movq    (sp)+,r2                    ;restore destination
              54    06     C0 0041          addl    #cvt_k_dst_vcha,r4          ;update case index for vcha dest
      00000000'GF   00     FB 0044          calls   #0,g^pli$cvrt_cg_r3         ;convert it
         0C AC      08     CA 004B          bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                           05 004F          rsb                                 ;return
                              0050
                              0050  ;pli$getebit_r6
                              0050  ;     inputs:
                              0050  ;        r0 - address of element to get
                              0050  ;        r1 - size/prec of element to get
                              0050  ;        r2 - offset of element to get
                              0050  ;        r11 - address of stream block
                              0050  ;        ap - address of file control block
                              0050  ;     outputs:
                              0050  ;        none
                              0050  ;     side effects:
                              0050  ;        r0-r6 are destroyed
                              0050  ;
                              0050
                              0050  pli$getebit_r6::
         0C AC      08     C8 0050          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
   0C AC    04000000 8F    CA 0054          bicl    #atr_m_flttrg,fcb_l_attr(ap) ;clr floating target flag
              7E    50     7D 005C          movq    r0,-(sp)                    ;save dest
                    52     DD 005F          pushl   r2                          ;save offset
         00000000'GF       16 0061          jsb     g^pli$$getfmt_r6            ;get format
                 56 8ED0      0067          popl    r6                          ;restore offset
              52    8E     7D 006A          movq    (sp)+,r2                    ;restore dest
              54    07     C0 006D          addl    #cvt_k_dst_bit,r4           ;update case index for bit dest
      00000000'GF   00     FB 0070          calls   #0,g^pli$cvrt_cg_r3         ;convert it
         0C AC      08     CA 0077          bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                           05 007B          rsb                                 ;return
                              007C
                              007C  ;pli$geteabit_r6
                              007C  ;     inputs:
                              007C  ;        r0 - address of element to get
                              007C  ;        r1 - size/prec of element to get
                              007C  ;        r11 - address of stream block
                              007C  ;        ap - address of file control block
                              007C  ;     outputs:
                              007C  ;        none
                              007C  ;     side effects:
                              007C  ;        r0-r6 are destroyed
                              007C  ;
                              007C
                              007C  pli$geteabit_r6::
         0C AC      08     C8 007C          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
   0C AC    04000000 8F    CA 0080          bicl    #atr_m_flttrg,fcb_l_attr(ap) ;clr floating target flag
              7E    50     7D 0088          movq    r0,-(sp)                    ;save destination
         00000000'GF       16 008B          jsb     g^pli$$getfmt_r6            ;get format
              52    8E     7D 0091          movq    (sp)+,r2                    ;restore destination
              54    08     C0 0094          addl    #cvt_k_dst_abit,r4          ;update case index for abit dest
```

```
       00000000'GF    00    FB  0097          calls   #0,g^pli$cvrt_cg_r3     ;convert it
             0C AC    08    CA  009E          bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                            05  00A2          rsb                             ;return
                                00A3
                                00A3  ;pli$getefixb_r6
                                00A3  ;     inputs:
                                00A3  ;        r0 - address of element to get
                                00A3  ;        r1 - size/prec of element to get
                                00A3  ;        r11 - address of stream block
                                00A3  ;        ap - address of file control block
                                00A3  ;     outputs:
                                00A3  ;        none
                                00A3  ;     side effects:
                                00A3  ;        r0-r6 are destroyed
                                00A3  ;
                                00A3
                                00A3  pli$getefixb_r6::
             0C AC    08    C8  00A3          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
   0C AC   04000000   8F    CA  00A7          bicl    #atr_m_flttrg,fcb_l_attr(ap) ;clr floating target flag
                7E    50    7D  00AF          movq    r0,-(sp)                ;save destination
         00000000'GF        16  00B2          jsb     g^pli$$getfmt_r6        ;get format
                52    8E    7D  00B8          movq    (sp)+,r2                ;restore destination
                54    01    C0  00BB          addl    #cvt_k_dst_fixb,r4      ;update case index for fixb dest
       00000000'GF    00    FB  00BE          calls   #0,g^pli$cvrt_cg_r3     ;convert it
             0C AC    08    CA  00C5          bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                            05  00C9          rsb                             ;return
                                00CA
                                00CA  ;pli$getefixd_r6
                                00CA  ;     inputs:
                                00CA  ;        r0 - address of element to get
                                00CA  ;        r1 - size/prec of element to get
                                00CA  ;        r11 - address of stream block
                                00CA  ;        ap - address of file control block
                                00CA  ;     outputs:
                                00CA  ;        none
                                00CA  ;     side effects:
                                00CA  ;        r0-r6 are destroyed
                                00CA  ;
                                00CA
                                00CA  pli$getefixd_r6::
             0C AC    08    C8  00CA          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
   0C AC   04000000   8F    CA  00CE          bicl    #atr_m_flttrg,fcb_l_attr(ap) ;clr floating target flag
                7E    50    7D  00D6          movq    r0,-(sp)                ;save destination
         00000000'GF        16  00D9          jsb     g^pli$$getfmt_r6        ;get format
                52    8E    7D  00DF          movq    (sp)+,r2                ;restore destination
                54    03    C0  00E2          addl    #cvt_k_dst_fixd,r4      ;update case index for fixd dest
       00000000'GF    00    FB  00E5          calls   #0,g^pli$cvrt_cg_r3     ;convert it
             0C AC    08    CA  00EC          bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                            05  00F0          rsb                             ;return
                                00F1
                                00F1  ;pli$geteflt b_r6
                                00F1  ;     inputs:
                                00F1  ;        r0 - address of element to get
                                00F1  ;        r1 - size/prec of element to get
                                00F1  ;        r11 - address of stream block
                                00F1  ;        ap - address of file control block
                                00F1  ;     outputs:
```

```
                                00F1  ;         none
                                00F1  ;     side effects:
                                00F1  ;         r0-r6 are destroyed
                                00F1  ;
                                00F1
                                00F1  pli$getefltb_r6::
        04000008 8F    C8       00F1          bisl    #<atr_m_recur!atr_m_flttrg>,- ;set recursion
              0C AC             00F7                  fcb_l_attr(ap)              ;and floating target
          08 51   07   E1       00F9          bbc     #7,r1,10$                   ;if g float
        0000008F 8F    DD       00FD          pushl   #<15+128>                   ;set max g float dec prec
                 0B    11       0103          brb     30$                         ;cont
            35   51    D1       0105  10$:    cmpl    r1,#53                      ;is it huge?
                 04    15       0108          bleq    20$                         ;if leq, no, cont
                 22    DD       010A          pushl   #34                         ;set max huge dec prec
                 02    11       010C          brb     30$                         ;cont
                 0F    DD       010E  20$:    pushl   #15                         ;set max d float dec prec
                                0110                                              ;(no need to bother w/ f/d since
                                0110                                              ;convert routines go to d)
            7E   50    7D       0110  30$:    movq    r0,-(sp)                    ;save destination
        00000000'GF    16       0113          jsb     g^pli$$getfmt_r6            ;get format
            52   8E    7D       0119          movq    (sp)+,r2                    ;restore destination
         5E   04 AE    9E       011C          movab   4(sp),sp                    ;clean dec prec from stack
            54   02    C0       0120          addl    #cvt_k_dst_fltb,r4          ;update case index for fltb dest
   00000000'GF   00    FB       0123          calls   #0,g^pli$cvrt_cg_r3         ;convert it
        04000008 8F    CA       012A          bicl    #<atr_m_recur!atr_m_flttrg>,- ;clr recursion flag
              0C AC             0130                  fcb_l_attr(ap)              ;and floating source
                       05       0132          rsb                                 ;return
                                0133
                                0133  ;pli$getefltd_r6
                                0133  ;     inputs:
                                0133  ;         r0 - address of element to get
                                0133  ;         r1 - size/prec of element to get
                                0133  ;         r11 - address of stream block
                                0133  ;         ap - address of file control block
                                0133  ;     outputs:
                                0133  ;         none
                                0133  ;     side effects:
                                0133  ;         r0-r6 are destroyed
                                0133  ;
                                0133
                                0133  pli$getefltd_r6::
        04000008 8F    C8       0133          bisl    #<atr_m_recur!atr_m_flttrg>,- ;set recursion
              0C AC             0139                  fcb_l_attr(ap)              ;and floating target
                 51    DD       013B          pushl   r1                          ;set float dec prec
            7E   50    7D       013D          movq    r0,-(sp)                    ;save destination
        00000000'GF    16       0140          jsb     g^pli$$getfmt_r6            ;get format
            52   8E    7D       0146          movq    (sp)+,r2                    ;restore destination
         5E   04 AE    9E       0149          movab   4(sp),sp                    ;clean dec prec from stack
            54   04    C0       014D          addl    #cvt_k_dst_fltd,r4          ;update case index for fltd dest
   00000000'GF   00    FB       0150          calls   #0,g^pli$cvrt_cg_r3         ;convert it
        04000008 8F    CA       0157          bicl    #<atr_m_recur!atr_m_flttrg>,- ;clr recursion flag
              0C AC             015D                  fcb_l_attr(ap)              ;and floating source
                       05       015F          rsb                                 ;return
                                0160
                                0160  ;pli$getepic_r6
                                0160  ;     inputs:
                                0160  ;         r0 - address of element to get
```

```
                                  0160  ;        r1 - size/prec of element to get
                                  0160  ;        r11 - address of stream block
                                  0160  ;        ap - address of file control block
                                  0160  ;     outputs:
                                  0160  ;        none
                                  0160  ;     side effects:
                                  0160  ;        r0-r6 are destroyed
                                  0160
                                  0160  pli$getepic_r6::
          0C AC      08       C8  0160         bisl     #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
   0C AC   04000000  8F       CA  0164         bicl     #atr_m_flttrg,fcb_l_attr(ap) ;clr floating target flag
                7E   50       7D  016C         movq     r0,-(sp)                    ;save destination
         00000000'GF 16           016F         jsb      g^pli$$getfmt_r6            ;get format
                52   8E       7D  0175         movq     (sp)+,r2                    ;restore destination
                54   00       C0  0178         addl     #cvt_k_dst_pic,r4           ;update case index for pic dest
      00000000'GF    00       FB  017B         calls    #0,g^pli$cvrt_cg_r3         ;convert it
          0C AC      08       CA  0182         bicl     #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                              05  0186         rsb                                  ;return
                                  0187
                                  0187         .end
```

```
ATR_M_FLTTRG                  = 04000000                  PLI$GETEVCHA_R6                 00000027 RG    02
ATR_M_RECUR                   = 00000008                  SIZ...                        = 00000001
CVT_K_DST_ABIT                = 00000008                  STK_L_AP                        00000008
CVT_K_DST_BIT                 = 00000007                  STK_L_ARG_LIST                  FFFFFFF8
CVT_K_DST_CHAR                = 00000005                  STK_L_CND_HND                   00000000
CVT_K_DST_FIXB                = 00000001                  STK_L_CND_LST                   FFFFFFF4
CVT_K_DST_FIXD                = 00000003                  STK_L_DISPLAY                   FFFFFFFC
CVT_K_DST_FLTB                = 00000002                  STK_L_FP                        0000000C
CVT_K_DST_FLTD                = 00000004                  STK_L_PC                        00000010
CVT_K_DST_PIC                 = 00000000                  STK_L_PSL                       00000004
CVT_K_DST_VCHA                = 00000006                  STK_L_REGS                      00000014
FCB_B_ENVIR                     000001C2                  STR_B_FIELD                     00000018
FCB_B_ESA                       0000012E                  STR_C_LEN                       00000C08
FCB_B_EXTRA                     0000003D                  STR_L_FLD_END                   00000014
FCB_B_FAB                       000000A6                  STR_L_FLD_PT                    00000010
FCB_B_IDENT                     00000040                  STR_L_FP                        00000004
FCB_B_IDENT_NAM                 00000042                  STR_L_FS                        0000000C
FCB_B_NAM                       000000F6                  STR_L_PARENT                    00000008
FCB_B_NUMKCBS                   0000003C                  STR_L_SP                        00000000
FCB_B_RAB                       00000062                  STR_L_STACK                     00000C04
FCB_C_LEN                       000001C2                  STR_L_STACK_END                 00000408
FCB_C_STRLEN                    00000034
FCB_L_ATTR                      0000000C
FCB_L_BUF                       00000014
FCB_L_BUF_END                   00000018
FCB_L_BUF_PT                    0000001C
FCB_L_CNDADDR                   000001B2
FCB_L_CONDIT                    000001AE
FCB_L_DTTR                      00000010
FCB_L_ERROR                     00000008
FCB_L_KCB                       00000038
FCB_L_NEXT                      00000000
FCB_L_PREVIOUS                  00000004
FCB_L_PRN                       00000034
FCB_Q_RFA                       00000020
FCB_W_COLUMN                    0000002E
FCB_W_IDENT_LEN                 00000040
FCB_W_LINE                      00000030
FCB_W_LINESIZE                  0000002A
FCB_W_PAGE                      00000032
FCB_W_PAGESIZE                  0000002C
FCB_W_REVISION                  00000028
GETOPT_B_BITS                   00000009
GETOPT_B_TMO                    00000008
GETOPT_C_LEN                    0000000A
GETOPT_L_FXDCTL                 00000000
GETOPT_L_PROMPT                 00000004
PLI$$GETFMT_R6                  ********   X    02
PLI$CVRT_CG_R3                  ********   X    02
PLI$GETEABIT_R6                 0000007C RG    02
PLI$GETEBIT_R6                  00000050 RG    02
PLI$GETECHAR_R6                 00000000 RG    02
PLI$GETEFIXB_R6                 000000A3 RG    02
PLI$GETEFIXD_R6                 000000CA RG    02
PLI$GETEFLTB_R6                 000000F1 RG    02
PLI$GETEFLTD_R6                 00000133 RG    02
PLI$GETEPIC_R6                  00000160 RG    02
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (   0.)  NOPIC    USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           FFFFFFFC  (     0.)  01 (   1.)  NOPIC    USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
_PLI$CODE                       00000187  (   391.)  02 (   2.)    PIC    USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC LONG

                                    +------------------------+
                                    . Performance indicators !
                                    +------------------------+

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                    9    00:00:00.05    00:00:00.35
Command processing               73    00:00:00.57    00:00:01.75
Pass 1                          188    00:00:06.92    00:00:16.09
Symbol table sort                 0    00:00:00.66    00:00:01.63
Pass 2                           56    00:00:01.30    00:00:02.91
Symbol table output              10    00:00:00.07    00:00:00.29
Psect synopsis output             1    00:00:00.03    00:00:00.03
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            337    00:00:09.60    00:00:23.06

The working set limit was 900 pages.
35926 bytes (71 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 574 non-local and 3 local symbols.
305 source lines were read in Pass 1, producing 11 object records in Pass 2.
20 pages of virtual memory were used to define 18 macros.

                                    +--------------------------+
                                    ! Macro library statistics !
                                    +--------------------------+

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1                7
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     7
TOTALS (all libraries)                               14

623 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=TRACEBACK/LIS=LIS$:PLIGETEDI/OBJ=OBJ$:PLIGETEDI MSRC$:PLIGETEDI/UPDATE=(ENH$:PLIGETEDI)+LIB$:PLIRTM
```

0308 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY