```
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
```

FILEID**PLIGETLIS

```
PPPPPPPP    LL          IIIIII     GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  LL          IIIIII     SSSSSSSS
PPPPPPPP    LL          IIIIII     GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  LL          IIIIII     SSSSSSSS
PP      PP  LL            II     GG          EE              TT      LL            II     SS
PP      PP  LL            II     GG          EE              TT      LL            II     SS
PP      PP  LL            II     GG          EE              TT      LL            II     SS
PP      PP  LL            II     GG          EE              TT      LL            II     SS
PPPPPPPP    LL            II     GG          EEEEEEEE        TT      LL            II       SSSSSS
PPPPPPPP    LL            II     GG          EEEEEEEE        TT      LL            II       SSSSSS
PP          LL            II     GG  GGGGGG  EE              TT      LL            II             SS
PP          LL            II     GG  GGGGGG  EE              TT      LL            II             SS
PP          LL            II     GG      GG  EE              TT      LL            II             SS
PP          LL            II     GG      GG  EE              TT      LL            II             SS
PP          LLLLLLLLLL  IIIIII     GGGGGG    EEEEEEEEEE      TT      LLLLLLLLLL  IIIIII     SSSSSSSS
PP          LLLLLLLLLL  IIIIII     GGGGGG    EEEEEEEEEE      TT      LLLLLLLLLL  IIIIII     SSSSSSSS


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     1          .title  pli$getlistitem
0000     2          .ident  /1-002/                                ; Edit WHM1002
0000     3  ;
0000     4  ;*****************************************************************************
0000     5  ;*                                                                           *
0000     6  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8  ;*  ALL RIGHTS RESERVED.                                                     *
0000     9  ;*                                                                           *
0000    10  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    11  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    12  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    13  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    14  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    15  ;*  TRANSFERRED.                                                             *
0000    16  ;*                                                                           *
0000    17  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    18  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    19  ;*  CORPORATION.                                                             *
0000    20  ;*                                                                           *
0000    21  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    22  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23  ;*                                                                           *
0000    24  ;*                                                                           *
0000    25  ;*****************************************************************************
0000    26  ;
0000    27  ;
0000    28
0000    29  ;++
0000    30  ; facility:
0000    31  ;
0000    32  ;       VAX/VMS PL1 runtime library
0000    33  ;
0000    34  ; abstract:
0000    35  ;
0000    36  ;       This module contains the pl1 runtime routines to get items from
0000    37  ;       pl1 stream file under list directed i/o.
0000    38  ;
0000    39  ;
0000    40  ; author: c. spitz 28-nov-79
0000    41  ;
0000    42  ; modified:
0000    43  ;
0000    44  ;
0000    45  ;       1-002   Bill Matthews   29-September-1982
0000    46  ;
0000    47  ;               Invoke macros $defdat and rtshare instead of $defopr and share.
0000    48  ;
0000    49  ;--
0000    50
0000    51  ;
0000    52  ; external definitions
0000    53  ;
0000    54          $deffcb                         ;define file control block
0000    55          $defstk                         ;define stack frame offsets
0000    56          $defstr                         ;define stream block offsets
0000    57          $defdat                         ;define operand node data types
```

```
                     0000     58          $defgetopt                      ;define get options block
                     0000     59          $defcvtind                      ;define convert indices
                     0000     60          $rabdef                         ;define rms rab offsets
                     0000     61          $rmsdef                         ;define rms error codes
                     0000     62
                     0000     63 ;
                     0000     64 ; local data
                     0000     65 ;
                     0000     66
                     0000     67          rtshare                 ;sharable
                     0000     68
                     0000     69
                     0000     70 ;++
                     0000     71 ; pli$getl****
                     0000     72 ;
                     0000     73 ; the pli$getl**** routines are called by the compiled code to get items
                     0000     74 ; from a stream input file under list directed transmission. each routine
                     0000     75 ; saves the target item descriptor, calls pli$$getnlis_r6 to get the
                     0000     76 ; next field, and then calls pli$cvrt_cg_r3 to store the item in the target.
                     0000     77 ;--
                     0000     78
                     0000     79
                     0000     80 ;pli$getlchar_r6
                     0000     81 ; inputs:
                     0000     82 ;        r0 - address of element to get
                     0000     83 ;        r1 - size/precision
                     0000     84 pli$getlchar_r6::
      0C AC    08    C8  0000     85          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
         7E    50    7D  0004     86          movq    r0,-(sp)                ;save destination
00000000'GF    16    0007     87          jsb     g^pli$$getnlis_r6       ;get input field
         52    8E    7D  000D     88          movq    (sp)+,r2                ;get destination back
               50    D5  0010     89          tstl    r0                      ;skip field?
               0A    13  0012     90          beql    20$                     ;if eql, then yes
               54    05  C0  0014     91          addl    #cvt_k_dst_char,r4      ;set for char dest
00000000'GF    00    FB  0017     92          calls   #0,g^pli$cvrt_cg_r3     ;convert field to dest
      0C AC    08    CA  001E     93 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
               05  0022     94          rsb                             ;return
                     0023     95
                     0023     96 ;pli$getlvcha_r6
                     0023     97 ; inputs:
                     0023     98 ;        r0 - address of element to get
                     0023     99 ;        r1 - size/precision
                     0023    100 pli$getlvcha_r6::
      0C AC    08    C8  0023    101          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
               60    B4  0027    102          clrw    (r0)                    ;clear length
         7E    50    7D  0029    103          movq    r0,-(sp)                ;save destination
00000000'GF    16    002C    104          jsb     g^pli$$getnlis_r6       ;get input field
         52    8E    7D  0032    105          movq    (sp)+,r2                ;get destination back
               50    D5  0035    106          tstl    r0                      ;skip field?
               0A    13  0037    107          beql    20$                     ;if eql, then yes
               54    06  C0  0039    108          addl    #cvt_k_dst_vcha,r4      ;set for vcha dest
00000000'GF    00    FB  003C    109          calls   #0,g^pli$cvrt_cg_r3     ;convert field to dest
      0C AC    08    CA  0043    110 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
               05  0047    111          rsb                             ;return
                     0048    112
                     0048    113 ;pli$getlfixb_r6
                     0048    114 ; inputs:
```

```
                        0048   115 ;       r0 - address of element to get
                        0048   116 ;       r1 - size/precision
                        0048   117 pli$getlfixb_r6::
            0C AC   08  C8  0048   118          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
               7E   50  7D  004C   119          movq    r0,-(sp)                    ;save destination
  00000000'GF       16  004F   120          jsb     g^pli$$getnlis_r6           ;get input field
               52   8E  7D  0055   121          movq    (sp)+,r2                    ;get destination back
                    50  D5  0058   122          tstl    r0                          ;skip field?
                    0A  13  005A   123          beql    20$                         ;if eql, then yes
               54   01  C0  005C   124          addl    #cvt_k_dst_fixb,r4          ;set for fixb dest
  00000000'GF       00  FB  005F   125          calls   #0,g^pli$cvrt_cg_r3         ;convert field to dest
            0C AC   08  CA  0066   126 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
                    05  006A   127          rsb                                 ;return
                        006B   128
                        006B   129 ;pli$getlfixd_r6
                        006B   130 ; inputs:
                        006B   131 ;       r0 - address of element to get
                        006B   132 ;       r1 - size/precision
                        006B   133 pli$getlfixd_r6::
            0C AC   08  C8  006B   134          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
               7E   50  7D  006F   135          movq    r0,-(sp)                    ;save destination
  00000000'GF       16  0072   136          jsb     g^pli$$getnlis_r6           ;get input field
               52   8E  7D  0078   137          movq    (sp)+,r2                    ;get destination back
                    50  D5  007B   138          tstl    r0                          ;skip field?
                    0A  13  007D   139          beql    20$                         ;if eql, then yes
               54   03  C0  007F   140          addl    #cvt_k_dst_fixd,r4          ;set for fixd dest
  00000000'GF       00  FB  0082   141          calls   #0,g^pli$cvrt_cg_r3         ;convert field to dest
            0C AC   08  CA  0089   142 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
                    05  008D   143          rsb                                 ;return
                        008E   144
                        008E   145 ;pli$getlbit_r6
                        008E   146 ; inputs:
                        008E   147 ;       r0 - address of element to get
                        008E   148 ;       r1 - size/precision
                        008E   149 ;       r2 - offset of target
                        008E   150 pli$getlbit_r6::
            0C AC   08  C8  008E   151          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
               7E   50  7D  0092   152          movq    r0,-(sp)                    ;save destination
                    52  DD  0095   153          pushl   r2                          ;save offset
  00000000'GF       16  0097   154          jsb     g^pli$$getnlis_r6           ;get input field
                    56 8ED0  009D   155          popl    r6                          ;set offset of dest
               52   8E  7D  00A0   156          movq    (sp)+,r2                    ;get destination back
                    50  D5  00A3   157          tstl    r0                          ;skip field?
                    0A  13  00A5   158          beql    20$                         ;if eql, then yes
               54   07  C0  00A7   159          addl    #cvt_k_dst_bit,r4           ;set for bit dest
  00000000'GF       00  FB  00AA   160          calls   #0,g^pli$cvrt_cg_r3         ;convert field to dest
            0C AC   08  CA  00B1   161 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
                    05  00B5   162          rsb                                 ;return
                        00B6   163
                        00B6   164 ;pli$getlabit_r6
                        00B6   165 ; inputs:
                        00B6   166 ;       - address of element to get
                        00B6   167 ;       - size/precision
                        00B6   168 pli$getlabit_r6::
            0C AC   08  C8  00B6   169          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
               7E   50  7D  00BA   170          movq    r0,-(sp)                    ;save destination
  00000000'GF       16  00BD   171          jsb     g^pli$$getnlis_r6           ;get input field
```

```
                52   8E   7D  00C3   172          movq    (sp)+,r2                ;get destination back
                     50   D5  00C6   173          tstl    r0                      ;skip field?
                     0A   13  00C8   174          beql    20$                     ;if eql, then yes
                54   08   C0  00CA   175          addl    #cvt_k_dst_abit,r4      ;set for abit dest
     00000000'GF     00   FB  00CD   176          calls   #0,g^pli$cvrt_cg_r3     ;convert field to dest
           0C AC     08   CA  00D4   177 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
                     05   00D8   178          rsb                             ;return
                          00D9   179
                          00D9   180 ;pli$getlfltb_r6
                          00D9   181 ; inputs:
                          00D9   182 ;        - address of element to get
                          00D9   183 ;        - size/precision
                          00D9   184 pli$getlfltb_r6::
           0C AC     08   C8  00D9   185          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
              7E     50   7D  00DD   186          movq    r0,-(sp)                ;save destination
     00000000'GF     16   00E0   187          jsb     g^pli$$getnlis_r6       ;get input field
                52   8E   7D  00E6   188          movq    (sp)+,r2                ;get destination back
                     50   D5  00E9   189          tstl    r0                      ;skip field?
                     0A   13  00EB   190          beql    20$                     ;if eql, then yes
                54   02   C0  00ED   191          addl    #cvt_k_dst_fltb,r4      ;set for fltb dest
     00000000'GF     00   FB  00F0   192          calls   #0,g^pli$cvrt_cg_r3     ;convert field to dest
           0C AC     08   CA  00F7   193 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
                     05   00FB   194          rsb                             ;return
                          00FC   195
                          00FC   196 ;pli$getlfltd_r6
                          00FC   197 ; inputs:
                          00FC   198 ;        - address of element to get
                          00FC   199 ;        - size/precision
                          00FC   200 pli$getlfltd_r6::
           0C AC     08   C8  00FC   201          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
              7E     50   7D  0100   202          movq    r0,-(sp)                ;save destination
     00000000'GF     16   0103   203          jsb     g^pli$$getnlis_r6       ;get input field
                52   8E   7D  0109   204          movq    (sp)+,r2                ;get destination back
                     50   D5  010C   205          tstl    r0                      ;skip field?
                     0A   13  010E   206          beql    20$                     ;if eql, then yes
                54   04   C0  0110   207          addl    #cvt_k_dst_fltd,r4      ;set for fltd dest
     00000000'GF     00   FB  0113   208          calls   #0,g^pli$cvrt_cg_r3     ;convert field to dest
           0C AC     08   CA  011A   209 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
                     05   011E   210          rsb                             ;return
                          011F   211
                          011F   212
                          011F   213 ;pli$getlpic_r6
                          011F   214 ; inputs:
                          011F   215 ;        - address of element to get
                          011F   216 ;        - size/precision
                          011F   217 pli$getlpic_r6::
           0C AC     08   C8  011F   218          bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursive flag
              7E     50   7D  0123   219          movq    r0,-(sp)                ;save destination
     00000000'GF     16   0126   220          jsb     g^pli$$getnlis_r6       ;get input field
                52   8E   7D  012C   221          movq    (sp)+,r2                ;get destination back
                     50   D5  012F   222          tstl    r0                      ;skip field?
                     0A   13  0131   223          beql    20$                     ;if eql, then yes
                54   00   C0  0133   224          addl    #cvt_k_dst_pic,r4       ;set for pic dest
     00000000'GF     00   FB  0136   225          calls   #0,g^pli$cvrt_cg_r3     ;convert field to dest
           0C AC     08   CA  013D   226 20$:     bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursive flag
                     05   0141   227          rsb                             ;return
                          0142   228
```

```
                0142    229             .end
```

```
ATR_M_RECUR                     = 00000008
CVT_K_DST_ABIT                  = 00000008
CVT_K_DST_BIT                   = 00000007
CVT_K_DST_CHAR                  = 00000005
CVT_K_DST_FIXB                  = 00000001
CVT_K_DST_FIXD                  = 00000003
CVT_K_DST_FLTB                  = 00000002
CVT_K_DST_FLTD                  = 00000004
CVT_K_DST_PIC                   = 00000000
CVT_K_DST_VCHA                  = 00000006
FCB_B_ENVIR                       000001C2
FCB_B_ESA                         0000012E
FCB_B_EXTRA                       0000003D
FCB_B_FAB                         000000A6
FCB_B_IDENT                       00000040
FCB_B_IDENT_NAM                   00000042
FCB_B_NAM                         000000F6
FCB_B_NUMKCBS                     0000003C
FCB_B_RAB                         00000062
FCB_C_LEN                         000001C2
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$$GETLIS_R6                    ********    X   02
PLI$CVRT_CG_R3                    ********    X   02
PLI$GETLABIT_R6                   000000B6 RG     02
PLI$GETLBIT_R6                    0000008E RG     02
PLI$GETLCHAR_R6                   00000000 RG     02
PLI$GETLFIXB_R6                   00000048 RG     02
PLI$GETLFIXD_R6                   0000006B RG     02
PLI$GETLFLTB_R6                   000000D9 RG     02
PLI$GETLFLTD_R6                   000000FC RG     02
PLI$GETLPIC_R6                    0000011F RG     02
PLI$GETLVCHA_R6                   00000023 RG     02

SIZ...                          = 00000001
STK_L_AP                          00000008
STK_L_ARG_LIST                    FFFFFFF8
STK_L_CND_HND                     00000000
STK_L_CND_LST                     FFFFFFF4
STK_L_DISPLAY                     FFFFFFFC
STK_L_FP                          0000000C
STK_L_PC                          00000010
STK_L_PSL                         00000004
STK_L_REGS                        00000014
STR_B_FIELD                       00000018
STR_C_LEN                         00000C08
STR_L_FLD_END                     00000014
STR_L_FLD_PT                      00000010
STR_L_FP                          00000004
STR_L_FS                          0000000C
STR_L_PARENT                      00000008
STR_L_SP                          00000000
STR_L_STACK                       00000C04
STR_L_STACK_END                   00000408
```

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	00000142 (322.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	10	00:00:00.05	00:00:00.64
Command processing	73	00:00:00.55	00:00:03.51
Pass 1	188	00:00:06.81	00:00:17.31
Symbol table sort	0	00:00:00.66	00:00:01.48
Pass 2	46	00:00:01.25	00:00:03.66
Symbol table output	8	00:00:00.07	00:00:00.07
Psect synopsis output	2	00:00:00.03	00:00:00.24
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	327	00:00:09.42	00:00:26.96

The working set limit was 900 pages.
35233 bytes (69 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 9 local symbols.
229 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
_255DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1	7
_255DUA28:[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$:PLIGETLIS/OBJ=OBJ$:PLIGETLIS MSRC$:PLIGETLIS/UPDATE=(ENH$:PLIGETLIS)+LIB$:PLIRTM
```

PLIFORMAT
LIS

PLIGETBUF
LIS

PLIMSGTXT
LIS

PLIGETEDI
LIS

PLIHEEP
LIS

PLIPUTFIL
LIS

PLIRMSBIS
LIS

PLIRECOPT
LIS

PLIOPEN
LIS

PLIREAD
LIS

PLIPROTEC
LIS

PLIREWRIT
LIS

PLIGETLIS
LIS

PLIPUTEDI
LIS

PLIPKDIVL
LIS

PLIPUTLIS
LIS

PLIMSGPTR
LIS

PLIPKDIVS
LIS

PLIPUTBUF
LIS

PLIGETFIL
LIS