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

```
                                                                  H  1
**FILE**ID**PLICONTRL









PPPPPPPP    LL            IIIIII      CCCCCCCC    OOOOOO    NN      NN  TTTTTTTTTT  RRRRRRRR    LL
PPPPPPPP    LL            IIIIII      CCCCCCCC    OOOOOO    NN      NN  TTTTTTTTTT  RRRRRRRR    LL
PP      PP  LL              II      CC          OO      OO  NN      NN      TT      RR      RR  LL
PP      PP  LL              II      CC          OO      OO  NN      NN      TT      RR      RR  LL
PP      PP  LL              II      CC          OO      OO  NNNN    NN      TT      RR      RR  LL
PP      PP  LL              II      CC          OO      OO  NNNN    NN      TT      RR      RR  LL
PPPPPPPP    LL              II      CC          OO      OO  NN  NN  NN      TT      RRRRRRRR    LL
PPPPPPPP    LL              II      CC          OO      OO  NN  NN  NN      TT      RRRRRRRR    LL
PP          LL              II      CC          OO      OO  NN    NNNN      TT      RR  RR      LL
PP          LL              II      CC          OO      OO  NN    NNNN      TT      RR  RR      LL
PP          LL              II      CC          OO      OO  NN      NN      TT      RR    RR    LL              ....
PP          LL              II      CC          OO      OO  NN      NN      TT      RR    RR    LL              ....
PP          LLLLLLLLLL    IIIIII      CCCCCCCC    OOOOOO    NN      NN      TT      RR      RR  LLLLLLLLLL      ....
PP          LLLLLLLLLL    IIIIII      CCCCCCCC    OOOOOO    NN      NN      TT      RR      RR  LLLLLLLLLL      ....

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
```

Table of contents

```
0000          .title  pli$control - pli runtime library control routines
0000          .ident  /1-004/                                  ; Edit LEB1004
0000                                                           ; Edit CGN1003
0000                                                           ; Edit WHM1002
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    ; facility:
0000    ;
0000    ;       VAX/VMS PLI  runtime library
0000    ;
0000    ; abstract:
0000    ;
0000    ; This module contains the data structures and code necessary to control
0000    ; the PL/I runtime library.
0000    ;
0000    ; author: R. Heinen 22-Jan-1980
0000    ;
0000    ; Modifications:
0000    ;
0000    ;
0000    ;       1-002   Bill Matthews   29-September-1982
0000    ;
0000    ;               Invoke macros $defdat and rtshare instead of $defopr and share.
0000    ;
0000    ;       1-003   Chip Nylander   23-February-1983
0000    ;
0000    ;               When closing all files for exit, call PLI$CLOSE with a
0000    ;               new optional parameter that tells it what severity level
0000    ;               an error should be signalled with.
0000    ;
0000    ;       1-004   Linda Benson    12-May-1983
0000    ;--
0000
0000    ;
```

```
                 0000  ; external definitions
                 0000  ;
                 0000          $defstk                         ; define runtine stack
                 0000          $stsdef                         ; sts$k symbols
                 0000
                 0000  ;
                 0000  ; global data
                 0000  ;
                 0000  ; runtime control block
                 0000  ;
             00000000          .psect pli$gl_rt_cblk,noshr,pic,usr,gbl,ovr,wrt,rd,noexe,rel
                 0000  pli$gl_rt_cblk:
        000000C0 0000          .blkl   48                      ; exit control block and runtime flags
                 00C0  pli$fcb_head::
        000000C8 00C0          .blkl   2
                 00C8
                 00C8  ;
                 00C8  ; define offsets into control block
                 00C8  ;
        00000000 00C8  cblk_l_exit_blk        = 0             ; exit control block
        00000004 00C8  cblk_l_exit_hnd        = 4             ; exit handler address
        00000008 00C8  cblk_l_exit_argc       = 8             ; exit arg count
        0000000C 00C8  cblk_l_exit_addr       = 12            ; address to store reason
        00000010 00C8  cblk_q_fcb_list        = 16            ; fcb list head
        00000018 00C8  cblk_l_exit_reason     = 24            ; reason for exit
        0000001C 00C8  cblk_l_flags           = 28            ; control flags
        00000000 00C8          cblk_v_inited  = 0             ; initialized
        00000001 00C8          cblk_v_main    = 1             ; options main
                 00C8
                 00C8          rtshare
```

```
                           0000          .sbttl  pli$optionsmain - initialize stack and options main program
                           0000  ;++
                           0000  ; pli$optionsmain - initialize stack and options main program
                           0000  ;
                           0000  ; functional description:
                           0000  ;
                           0000  ; This routine initializes the stack frame for options main procedures
                           0000  ; and declares an exit handler for for the program.
                           0000  ;
                           0000  ; inputs:
                           0000  ;
                           0000  ;       none
                           0000  ;
                           0000  ; outputs:
                           0000  ;
                           0000  ;       none
                           0000  ;--
                           0000  pli$optionsmain::
            F4 AD      D4  0000          clrl    stk_l_cnd_lst(fp)       ; set up control block list
6D   00000000'GF      9E  0003          movab   g^pli$optmain_hnd,(fp)  ; set up frame handler
               0D      10  000A          bsbb    initialize_exithandler  ; initialize exit handler
   07 1C A2    01      E3  000C          bbcs    #cblk_v_main,cblk_l_flags(r2),10$; br if already initialized
6D   00000000'GF      9E  0011          movab   g^pli$def_hnd,(fp)      ; setup normal condition handler
                       05  0018  10$:    rsb
                           0019  ;
                           0019  ; check for uninitialized exit handler
                           0019  ;
                           0019  initialize_exithandler:
52   00000000'GF      9E  0019          movab   g^pli$gl_rt_cblk,r2     ; address control block
   28 1C A2    00      E0  0020          bbs     #cblk_v_inited,cblk_l_flags(r2),10$; br if inited
                           0025  ;
                           0025  ; setup control block
                           0025  ;
      1C A2    01      88  0025          bisb    #1@cblk_v_inited,cblk_l_flags(r2); set inited
   10 A2    10 A2      9E  0029          movab   cblk_q_fcb_list(r2),cblk_q_fcb_list(r2); initialize list head
   14 A2    10 A2      9E  002E          movab   cblk_q_fcb_list(r2),cblk_q_fcb_list+4(r2);
               62      D4  0033          clrl    cblk_l_exit_blk(r2)     ; build exit handler control block
 04 A2   0064'CF      9E  0035          movab   w^pli$exit_hnd,cblk_l_exit_hnd(r2);
      08 A2    05      D0  003B          movl    #5,cblk_l_exit_argc(r2) ;
   0C A2    18 A2      9E  003F          movab   cblk_l_exit_reason(r2),cblk_l_exit_addr(r2);
                           0044          $dclexh_s       cblk_l_exit_blk(r2)
                       05  004D  10$:    rsb                             ; done
```

```
                             004E  ;++
                             004E  ; pli$$term_prog - set up termination
                             004E  ;
                             004E  ; functional description:
                             004E  ;
                             004E  ; This routine is used prior to a resignal to the command interpreter
                             004E  ; following a ERROR condition that was not handled.  The action is to reset
                             004E  ; the options_main flag to prevent the FINISH condition from being signalled.
                             004E  ; When an exit is done inside the command interpreter, the FINISH signal
                             004E  ; would not be fielded by any existing stack frame.  This action prevents
                             004E  ; a junk FINISH condition message from being displayed.
                             004E  ;
                             004E  ; inputs:
                             004E  ;
                             004E  ;       none.
                             004E  ;
                             004E  ; outputs:
                             004E  ;
                             004E  ;       none.
                             004E  ;--
                             004E pli$$term_prog::
50   0000001C'GF       9E  004E          movab   g^pli$gl_rt_cblk+cblk_l_flags,r0; address area
            60   02    8A  0055          bicb    #1@cblk_v_main,(r0)              ; reset options_main
                       05  0058          rsb
```

```
                  0059           .sbttl  pli$link_fcb - link fcb onto exit handler chain
                  0059  ;++
                  0059  ; pli$link_fcb
                  0059  ;
                  0059  ; functional description:
                  0059  ;
                  0059  ; This routine links an fcb onto the exit handler list in pli$gl_rt_cblk.
                  0059  ; If the entry is the first entry then the control block is designated as
                  0059  ; an exit handler control block.
                  0059  ;
                  0059  ; inputs:
                  0059  ;
                  0059  ;        r6 = fcb address to link
                  0059  ;
                  0059  ; outputs:
                  0059  ;
                  0059  ;        none.
                  0059  ;--
                  0059  pli$link_fcb::
         07   BB  0059           pushr   #^m<r0,r1,r2>
         BC   10  005B           bsbb    initialize_exithandler  ; initialize handler and block
                  005D  ;
                  005D  ; link fcb onto list
                  005D  ;
10 A2    66   0E  005D  10$:     insque  (r6),cblk_q_fcb_list(r2); insert fcb in list
         07   BA  0061           popr    #^m<r0,r1,r2>
              05  0063           rsb
```

```
                                 0064          .sbttl  pli$exit_hnd - exit handler for PL/I runtime
                                 0064    ;++
                                 0064    ; pli$exit_hnd - PL/I exit handler
                                 0064    ;
                                 0064    ; functional description:
                                 0064    ;
                                 0064    ; This routine is entered on exit in order to clean up the I/O database.
                                 0064    ; Prior to cleaning up the I/O Pli$_finish is signalled.
                                 0064    ;
                                 0064    ; inputs:
                                 0064    ;
                                 0064    ;      0(ap) = 1
                                 0064    ;      4(ap) = address of the exit control block argument list
                                 0064    ;
                                 0064    ; outputs:
                                 0064    ;
                                 0064    ;      none.
                                 0064    ;
                                 0064    ;--
                           0000  0064          .entry  pli$exit_hnd,0
                                 0066    ;
                                 0066    ; give finish condition handlers a chance
                                 0066    ;
      0D 14 AC     01       E1   0066          bbc     #cblk_v_main,cblk_l_flags-cblk_l_exit_argc(ap),10$; skip signal if n
         00000000'8F        DD   006B          pushl   #pli$_finish              ; signal finish condition
     00000000'GF   01       FB   0071          calls   #1,g^lib$signal           ;
                                 0078    ;
                                 0078    ; close all open files
                                 0078    ;
                   02       DD   0078    10$:  pushl   #sts$k_error              ; signal E-level errors, not F-level
                   00       DD   007A          pushl   #0                        ; no environment option
           7E   08 BC       0F   007C          remque  @cblk_q_fcb_list-cblk_l_exit_argc(ap),-(sp); get a control block
                   09       1D   0080          bvs     15$                       ; if vset then done
     00000000'GF   03       FB   0082          calls   #3,g^pli$close            ; close file
                   ED       11   0089          brb     10$                       ; continue until done
                            04   008B    15$:  ret
```

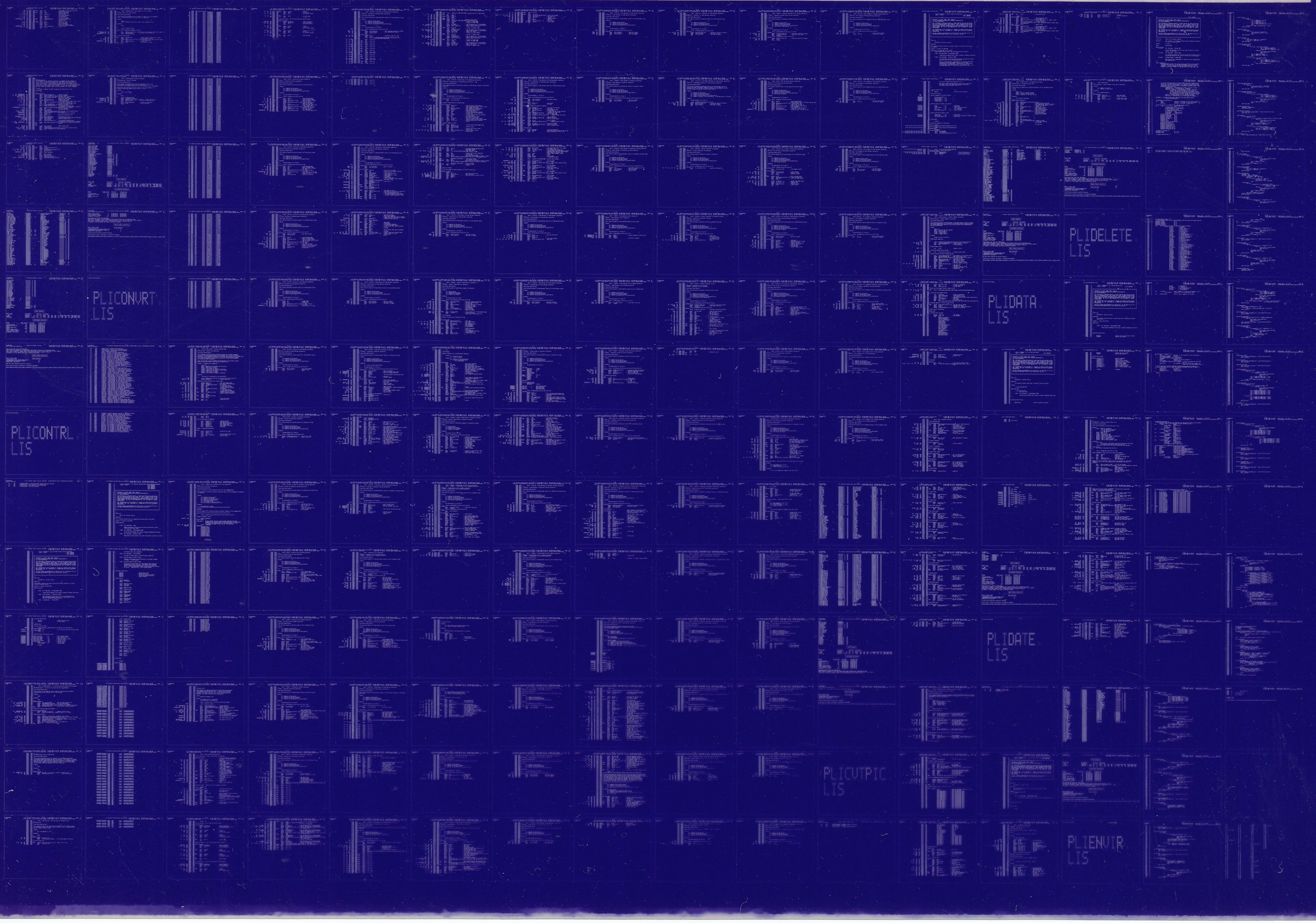
```
                                 008C          .sbttl  pli$stop_prog - stop pl1 program
                                 008C  ;++
                                 008C  ; pli$stop_prog - stop pl1 program
                                 008C  ;
                                 008C  ; functional description:
                                 008C  ;
                                 008C  ; This routine is envoked when a RET or BLOCK_END is done in an
                                 008C  ; options main procedure or when a STOP is executed.
                                 008C  ; The action is to signal the finish condition and terminate the program.
                                 008C  ;
                                 008C  ; Note that this can cause infinite loops in the user program if it
                                 008C  ; handles FINISH and somehow does a STOP or ret etc.
                                 008C  ;
                                 008C  ; inputs:
                                 008C  ;
                                 008C          4(ap) = exit status
                                 008C  ;
                                 008C  ; outputs:
                                 008C  ;
                                 008C          none.
                                 008C  ;--
                           0000  008C          .entry  pli$stop_prog,0
50     00000000'GF          9E  008E          movab   g^pli$gl_rt_cblk,r0
        1C A0    02          8A  0095          bicb    #1@cblk_v_main,cblk_l_flags(r0) ; disable force exit
       00000000'8F          DD  0099          pushl   #pli$_finish            ; finish is the condition
                 50          7C  009F          clrq    r0                      ;
00000000'GF       01          FB  00A1          calls   #1,g^lib$signal         ; signal the condition
                                 00A8          $exit_s 4(ap)                   ; end the program
                                 00B2          .end
```

```
CBLK_L_EXIT_ADDR               = 0000000C
CBLK_L_EXIT_ARGC               = 00000008
CBLK_L_EXIT_BLK                = 00000000
CBLK_L_EXIT_HND                = 00000004
CBLK_L_EXIT_REASON             = 00000018
CBLK_L_FLAGS                   = 0000001C
CBLK_Q_FCB_LIST                = 00000010
CBLK_V_INITED                  = 00000000
CBLK_V_MAIN                    = 00000001
INITIALIZE_EXITHANDLER           00000019 R      03
LIB$SIGNAL                       ********    X   03
PLI$$TERM_PROG                   0000004E RG     03
PLI$CLOSE                        ********    X   03
PLI$DEF_HND                      ********    X   03
PLI$EXIT_HND                     00000064 RG     03
PLI$FCB_READ                     000000C0 RG     02
PLI$GL_RT_CBLK                   00000000 R      02
PLI$LINK_FCB                     00000059 RG     03
PLI$OPTIONSMAIN                  00000000 RG     03
PLI$OPTMAIN_HND                  ********    X   03
PLI$STOP_PROG                    0000008C RG     03
PLI$_FINISH                      ********    X   03
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
STS$K_ERROR                    = 00000002
SYS$DCLEXH                       ********   GX   03
SYS$EXIT                         ********   GX   03


                                                 +-----------------+
                                                 ! 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$GL_RT_CBLK                 000000C8  (   200.)   02 (   2.)    PIC    USR    OVR    REL    GBL NOSHR NOEXE   RD     WRT NOVEC BYTE
_PLI$CODE                      000000B2  (   178.)   03 (   3.)    PIC    USR    CON    REL    LCL   SHR   EXE   RD   NOWRT NOVEC LONG


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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   14     00:00:00.06    00:00:02.25
Command processing               87     00:00:00.46    00:00:05.42
Pass 1                           87     00:00:01.52    00:00:09.20
Symbol table sort                 0     00:00:00.04    00:00:00.04
Pass 2                           47     00:00:00.52    00:00:02.18
```

```
Symbol table output            4     00:00:00.04     00:00:00.04
Psect synopsis output          2     00:00:00.02     00:00:00.02
Cross-reference output         0     00:00:00.00     00:00:00.00
Assembler run totals         241     00:00:02.66     00:00:19.15

The working set limit was 900 pages.
6337 bytes (13 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 65 non-local and 5 local symbols.
250 source lines were read in Pass 1, producing 18 object records in Pass 2.
13 pages of virtual memory were used to define 12 macros.

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

Macro library name                              Macros defined
------------------                              --------------

_$255$DUA28:[PLIRTL.OBJ]PLIR,MAC.MLB;1                2
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     7
TOTALS (all libraries)                                9

126 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

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

0307 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



PLIDELETE
LIS

PLICONVRT
LIS

PLIDATA
LIS

PLICONTRL
LIS

PLIDATE
LIS

PLICUTPIC
LIS

PLIENVIR
LIS