

PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLL IIIII
LLLLLLLLLL IIIII
SSSSSSSS
SSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSS
SSSSSSSS

```


PLI\$CONTROL
Table of contents

- pli runtime library control routines ^{I 1} 16-SEP-1984 02:13:50 VAX/VMS Macro V04-00

Page 0

(1)	89	pli\$optionsmain - initialize stack and options main program
(1)	157	pli\$link_fcb - link fcb onto exit handler chain
(1)	185	pli\$exit_hnd - exit handler for PL/I runtime
(2)	222	pli\$stop_prog - stop pli program

PLI
1-0

```

0000 1      .title  pli$control - pli runtime library control routines
0000 2      .ident  /1-004/                                     ; Edit LEB1004
0000 3                                                    ; Edit CGN1003
0000 4                                                    ; Edit WHM1002
0000 5
0000 6 :*****
0000 7 :*
0000 8 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 :*  ALL RIGHTS RESERVED.
0000 11 :*
0000 12 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 :*  TRANSFERRED.
0000 18 :*
0000 19 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 :*  CORPORATION.
0000 22 :*
0000 23 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 :*
0000 26 :*
0000 27 :*****
0000 28
0000 29 ++
0000 30 : facility:
0000 31
0000 32 :     VAX/VMS PLI  runtime library
0000 33
0000 34 : abstract:
0000 35
0000 36 : This module contains the data structures and code necessary to control
0000 37 : the PL/I runtime library.
0000 38
0000 39 : author: R. Heinen 22-Jan-1980
0000 40
0000 41 : Modifications:
0000 42
0000 43
0000 44 :     1-002   Bill Matthews   29-September-1982
0000 45
0000 46 :             Invoke macros $defdat and rtshare instead of $defopr and share.
0000 47
0000 48 :     1-003   Chip Nylander   23-February-1983
0000 49
0000 50 :             When closing all files for exit, call PLI$CLOSE with a
0000 51 :             new optional parameter that tells it what severity level
0000 52 :             an error should be signalled with.
0000 53
0000 54 :     1-004   Linda Benson    12-May-1983
0000 55 : --
0000 56
0000 57 :

```



```
0000 58 ; external definitions
0000 59 ;
0000 60 $defstk ; define runtime stack
0000 61 $stsdef ; sts$ symbols
0000 62
0000 63 ;
0000 64 ; global data
0000 65 ;
0000 66 ; runtime control block
0000 67 ;
00000000 68 .psect pli$gl_rt_cblk,noshr,pic,usr,gbl,ovr,wrt,rd,noexe,rel
000000C0 0000 69 pli$gl_rt_cblk:
000000C0 0000 70 .blkl 48 ; exit control block and runtime flags
000000C8 00C0 71 pli$fcbl_head::
000000C8 00C0 72 .blkl 2
000000C8 00C8 73
000000C8 00C8 74 ;
000000C8 00C8 75 ; define offsets into control block
000000C8 00C8 76 ;
00000000 00C8 77 cblk_l_exit_blk = 0 ; exit control block
00000004 00C8 78 cblk_l_exit_hnd = 4 ; exit handler address
00000008 00C8 79 cblk_l_exit_argc = 8 ; exit arg count
0000000C 00C8 80 cblk_l_exit_addr = 12 ; address to store reason
00000010 00C8 81 cblk_q_fcb_list = 16 ; fcb list head
00000018 00C8 82 cblk_l_exit_reason = 24 ; reason for exit
0000001C 00C8 83 cblk_l_flags = 28 ; control flags
00000000 00C8 84 cblk_v_initd = 0 ; initialized
00000000 00C8 85 cblk_v_main = 1 ; options main
00000001 00C8 86
00000001 00C8 87 rtshare
```

```
0000 89 .sbttl pli$optionsmain - initialize stack and options main program
0000 90 :++
0000 91 : pli$optionsmain - initialize stack and options main program
0000 92 :
0000 93 : functional description:
0000 94 :
0000 95 : This routine initializes the stack frame for options main procedures
0000 96 : and declares an exit handler for for the program.
0000 97 :
0000 98 : inputs:
0000 99 :
0000 100 : none
0000 101 :
0000 102 : outputs:
0000 103 :
0000 104 : none
0000 105 :--
0000 106 pli$optionsmain::
6D 00000000'GF 04 0000 107 clrl stk_l_cnd_lst(fp) ; set up control block list
07 1C A2 01 9E 0003 108 movab g^pli$optmain_hnd,(fp) ; set up frame handler
6D 00000000'GF 0D 10 000A 109 bsbb initialize_exithandler ; initialize exit handler
07 1C A2 01 E3 000C 110 bbcs #cblk_v_main,cblk_l_flags(r2),10$; br if already initialized
6D 00000000'GF 9E 0011 111 movab g^pli$def_hnd,(fp) ; setup normal condition handler
05 0018 112 10$: rsb
0019 113 :
0019 114 : check for uninitialized exit handler
0019 115 :
0019 116 initialize_exithandler:
52 00000000'GF 9E 0019 117 movab g^pli$gl_rt_cblk,r2 ; address control block
28 1C A2 00 E0 0020 118 bbs #cblk_v_initd,cblk_l_flags(r2),10$; br if initd
0025 119 :
0025 120 : setup control block
0025 121 :
0025 122 bisb #1@cblk_v_initd,cblk_l_flags(r2); set initd
10 A2 10 A2 9E 0029 123 movab cblk_q_tcb_list(r2),cblk_q_tcb_list(r2); initialize list head
14 A2 10 A2 9E 002E 124 movab cblk_q_tcb_list(r2),cblk_q_tcb_list+4(r2);
04 A2 0064'CF 04 0033 125 clrl cblk_l_exit_blk(r2) ; build exit handler control block
08 A2 05 D0 0035 126 movab w^pli$exit_hnd,cblk_l_exit_hnd(r2);
0C A2 18 A2 9E 003B 127 movl #5,cblk_l_exit_argc(r2);
05 0044 128 movab cblk_l_exit_reason(r2),cblk_l_exit_addr(r2);
05 004D 129 $dclexh_s cblk_l_exit_blk(r2)
130 10$: rsb ; done
```



```
004E 132 :++
004E 133 : pli$$term_prog - set up termination
004E 134 :
004E 135 : functional description:
004E 136 :
004E 137 : This routine is used prior to a resignal to the command interpreter
004E 138 : following a ERROR condition that was not handled. The action is to reset
004E 139 : the options_main flag to prevent the FINISH condition from being signalled.
004E 140 : When an exit is done inside the command interpreter, the FINISH signal
004E 141 : would not be fielded by any existing stack frame. This action prevents
004E 142 : a junk FINISH condition message from being displayed.
004E 143 :
004E 144 : inputs:
004E 145 :
004E 146 : none.
004E 147 :
004E 148 : outputs:
004E 149 :
004E 150 : none.
004E 151 :--
004E 152 pli$$term_prog::
50 0000001C'GF 9E 004E 153 movab g^pli$gl_rt_cblk+cblk_l_flags,r0; address area
60 02 8A 0055 154 bicb #1@cblk_v_main,(r0) ; reset options_main
05 0058 155 rsb
```

```

0059 157 .sbttl pli$link_fcb - link fcb onto exit handler chain
0059 158 :++
0059 159 : pli$link_fcb
0059 160 :
0059 161 : functional description:
0059 162 :
0059 163 : This routine links an fcb onto the exit handler list in pli$gl_rt_cblk.
0059 164 : If the entry is the first entry then the control block is designated as
0059 165 : an exit handler control block.
0059 166 :
0059 167 : inputs:
0059 168 :
0059 169 :     r6 = fcb address to link
0059 170 :
0059 171 : outputs:
0059 172 :
0059 173 :     none.
0059 174 : --
0059 175 pli$link_fcb::
07  BB 0059 176     pushr    #^m<r0,r1,r2>
BC  10 005B 177     bsbb     initialize_exithandler ; initialize handler and block
005D 178 :
005D 179 : Link fcb onto list
005D 180 :
10 A2 66 0E 005D 181 10$: insque (r6),cblk_q_fcb_list(r2); insert fcb in list
07 07 0061 182     popr     #^m<r0,r1,r2>
05 0063 183     rsb

```



```
0064 185      .sbtll pli$exit_hnd - exit handler for PL/I runtime
0064 186      :++
0064 187      : pli$exit_hnd - PL/I exit handler
0064 188      :
0064 189      : functional description:
0064 190      :
0064 191      : This routine is entered on exit in order to clean up the I/O database.
0064 192      : Prior to cleaning up the I/O Plis_finish is signalled.
0064 193      :
0064 194      : inputs:
0064 195      :
0064 196      :     0(ap) = 1
0064 197      :     4(ap) = address of the exit control block argument list
0064 198      :
0064 199      : outputs:
0064 200      :
0064 201      :     none.
0064 202      :
0000 0064 203      :--
0064 204      : .entry pli$exit_hnd,0
0066 205      :
0066 206      : give finish condition handlers a chance
0066 207      :
0066 208      :     bbc      #cblk_v_main,cblk_l_flags-cblk_l_exit_argc(ap),10$; skip signal if n
00000000'GF 01 01 0066 209      :     pushl    #pli$finish      ; signal finish condition
0066 210      :     calls   #1,g^lib$signal      ;
0066 211      :
0066 212      : close all open files
0066 213      :
0066 214 10$:      :     pushl    #sts$k_error      ; signal E-level errors, not F-level
0066 215      :     pushl    #0      ; no environment option
0066 216      :     remque   @cblk_q_fcb_list-cblk_l_exit_argc(ap),-(sp); get a control block
0066 217      :     bvs      15$      ; if vset then done
0066 218      :     calls   #3,g^pli$close      ; close file
0066 219      :     brb      10$      ; continue until done
0066 220 15$:      :     ret
0066 220 15$:      :     ret
```

```
008C 222      .sbtcl pli$stop_prog - stop pl1 program
008C 223      :++
008C 224      : pli$stop_prog - stop pl1 program
008C 225      :
008C 226      : functional description:
008C 227      :
008C 228      : This routine is invoked when a RET or BLOCK_END is done in an
008C 229      : options main procedure or when a STOP is executed.
008C 230      : The action is to signal the finish condition and terminate the program.
008C 231      :
008C 232      : Note that this can cause infinite loops in the user program if it
008C 233      : handles FINISH and somehow does a STOP or ret etc.
008C 234      :
008C 235      : inputs:
008C 236      :
008C 237      :      4(ap) = exit status
008C 238      :
008C 239      : outputs:
008C 240      :
008C 241      :      none.
008C 242      :--
008C 243      .entry pli$stop_prog,0
50 00000000'GF 0000 008E 244      movab g^pli$gl_rt_cblk,r0
      1C A0 02 8A 0095 245      bicb #1acblk_v_main,cblk_l_flags(r0) ; disable force exit
      00000000'8F DD 0099 246      pushl #pli$finish ; finish is the condition
      50 7C 009F 247      clrq r0 ;
00000000'GF 01 FB 00A1 248      calls #1,g^lib$signal ; signal the condition
      00A8 249      $exit_s 4(ap) ; end the program
00B2 250      .end
```


PLISCONTROL
Symbol table

- pli runtime library control routines

16-SEP-1984 02:13:50
6-SEP-1984 11:36:41

VAX/VMS Macro V04-00
[PLIRTL.SRC]PLICONTROL.MAR;1

Page 8
(2)

```

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
LIBSSIGNAL            ***** X      03
PLIS$TERM_PROG        0000004E RG     03
PLIS$CLOSE            ***** X      03
PLIS$DEF_HND          ***** X      03
PLIS$EXIT_HND         00000064 RG     03
PLIS$FCB_HEAD         000000C0 RG     02
PLIS$GL_RT_CBLK       00000000 R      02
PLIS$LINK_FCB         00000059 RG     03
PLIS$OPTIONSMAIN      00000000 RG     03
PLIS$OPTMAIN_HND      ***** X      03
PLIS$STOP_PROG        0000008C RG     03
PLIS$FINISH           ***** X      03
STK_L_AP              00000008
STK_L_ARG_LIST        FFFFFFFF8
STK_L_CND_HND         00000000
STK_L_CND_LST         FFFFFFFF4
STK_L_DISPLAY         FFFFFFFFC
STK_L_FP              0000000C
STK_L_PC              00000010
STK_L_PSL             00000004
STK_L_REGS            00000014
ST$K_ERROR            = 00000002
SY$SDCLEXH            ***** GX     03
SY$SEXIT              ***** 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
\$ABSS	FFFFFFFC (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
PLIS\$GL_RT_CBLK	000000C8 (200.)	02 (2.)	PIC USR OVR REL GBL NOSHR NOEXE RD WRT NOVEC BYTE
_PLISCODE	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]PLIRTMAC.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=LISS: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

