


```

PPPPPPP LL      IIIIII  CCCCCCCC LL      000000  SSSSSSSS EEEEEEEEE
PPPPPPP LL      IIIIII  CCCCCCCC LL      000000  SSSSSSSS EEEEEEEEE
PP      PP LL      II      CC      LL      00      00  SS      EE
PP      PP LL      II      CC      LL      00      00  SS      EE
PP      PP LL      II      CC      LL      00      00  SS      EE
PP      PP LL      II      CC      LL      00      00  SS      EE
PPPPPPP LL      II      CC      LL      00      00  SSSSSS EEEEEEE
PPPPPPP LL      II      CC      LL      00      00  SSSSSS EEEEEEE
PP      LL      II      CC      LL      00      00      SS  EE
PP      LL      II      CC      LL      00      00      SS  EE
PP      LL      II      CC      LL      00      00      SS  EE
PP      LL      II      CC      LL      00      00      SS  EE
PP      LL      IIIIII  CCCCCCCC LLLLLLLLLL 000000  SSSSSSSS EEEEEEEEE
PP      LLLLLLLLLL IIIIII  CCCCCCCC LLLLLLLLLL 000000  SSSSSSSS EEEEEEEEE

```

```

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      IIIIII  SSSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS

```

```

0000 1      .title pl1$close - pl1 runtime close file
0000 2      .ident /1-004/
0000 3
0000 4
0000 5
0000 6
0000 7
0000 8
0000 9
0000 10     *
0000 11     * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 12     * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 13     * ALL RIGHTS RESERVED.
0000 14     *
0000 15     * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 16     * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 17     * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 18     * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 19     * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 20     * TRANSFERRED.
0000 21     *
0000 22     * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 23     * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 24     * CORPORATION.
0000 25     *
0000 26     * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 27     * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 28     *
0000 29     *
0000 30
0000 31
0000 32     ++
0000 33     facility:
0000 34
0000 35         VAX/VMS PL1 runtime library.
0000 36     abstract:
0000 37
0000 38         This module contains the pl1 runtime routine for closing a file.
0000 39
0000 40     author: c. spitz 18-jul-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 $defopr and share.
0000 48
0000 49         1-003   Chip Nylander   23-February-1983
0000 50
0000 51         Add new optional parameter to PL1$CLOSE. If parameter is
0000 52         present, it specifies the severity level at which to signal
0000 53         errors.
0000 54
0000 55         1-004   Dave Blickstein 26-March-1984
0000 56
0000 57         Change calls to LIB$FREE_VM to specify only the actual number

```

OC

```
0000 58 :  
0000 59 :  
0000 60 :  
0000 61 :  
0000 62 :--  
0000 63 :  
0000 64 :  
0000 65 :  
0000 66 : external definitions  
0000 67 :  
0000 68 :  
0000 69 $deffcb ;define file control block offsets  
0000 70 $defpl1rtcons ;define pl1 runtime constants  
0000 71 $defkcb ;define key control block offsets  
0000 72 $defenv ;define environment block offsets  
0000 73 $fabdef ;define fab offsets  
0000 74 $namdef ;define nam offsets  
0000 75 :  
0000 76 :  
0000 77 : local data  
0000 78 :  
0000 79 :  
0000 80 rtshare ;sharable  
0000 81 :  
0000 82 :
```

of arguments they push (2) to the CALLS instruction. The argument count was being used to pop local stack temporaries. This broke when new versions of the VM routines that took optional arguments were called.

```

0000 84
0000 85 :++
0000 86 : plisclose -- close a file
0000 87 :
0000 88 : functional description:
0000 89 :
0000 90 : This routine closes a pl1 file. The syntax of a close statement is:
0000 91 : CLOSE FILE(file-ref);
0000 92 :
0000 93 : inputs:
0000 94 : (ap) = number of arguments (1)
0000 95 : 4(ap) = address of the fcb for this file
0000 96 : 8(ap) = address of the environment option
0000 97 : 12(ap) = severity level at which to signal errors
0000 98 :
0000 99 : outputs:
0000 100 : the fcb is updated to reflect the closed status of the file.
0000 101 : an rms close is issued, the file block is deallocated, and the
0000 102 : fcb is removed from the open fcb list.
0000 103 :--
0000 104 :
0000 105 : .entry plisclose, ^m<r2,r3,r4,r5,r6,r7>
0002 106 :
0002 107 : check argument
0002 108 :
0002 109 : tstl (ap) ;if no parameters
0004 110 : bgr 10$ ;then
0006 111 : clrl r1 ;indicate no fcb
0008 112 : movl #plis_parm,r0 ;set invalid parameters
000F 113 : brw 50$ ;and fail
0012 114 :
0012 115 : if file opened, close it.
0012 116 :
0012 117 10$: movl 4(ap),r2 ;get address of fcb
0016 118 : bbs #atr_v_opened,fcb_l_attr(r2),20$ ;if file opened
001B 119 : brw 110$ ;then
001E 120 20$: bicl #<atr_m_opened!atr_m_bfall>,- ;set file not opened and
0024 121 : fcb_l_attr(r2) ;buffer not allocated
0026 122 : movab fcb_b_fab(r2),r3 ;get address of fab
002B 123 : bitl #<atr_m_record! - ;if record
0033 124 : atr_m_input>,fcb_l_attr(r2) ;or input
0033 125 : bneq 30$ ;then cont
0035 126 : bbc #atr_v_print,fcb_l_attr(r2),25$ ;if not print, cont
003A 127 : tstl fcb_l_prn(r2) ;any carriage control pending?
003F 128 : bneq 26$ ;if neq, yes, print it
003F 129 25$: cmpl fcb_l_buf(r2),fcb_l_buf_pt(r2) ;if buf empty
0044 130 : beql 30$ ;then cont
0046 131 26$: oushl ap ;save ap
0048 132 : movl r2,ap ;set fcb addr in ap
004B 133 : jsb g^plis$put_rec ;put the last record
0051 134 : popl ap ;restore ap
0054 135 30$: cmpl (ap),#2 ;close environment?
0057 136 : blss 37$ ;if lss, no
0059 137 : tstl 8(ap) ;anything there?
005C 138 : beql 37$ ;if eql, no
005E 139 : subl #env_k_len,sp ;get room for env block
0061 140 : movq r2,r6 ;save fcb and fab addr

```

```

6E 22 00 6E 00 2C 0064 141 movc5 #0,(sp),#0,#env_k_len,(sp) ;zero env block
      52 56 7D 006A 142 movq r6,r2 ;restore fcb and fab
6E 00000040 8F C8 006D 143 bisl #env_m_close,env_l_status(sp) ;set close status
      SE DD 0074 144 pushl sp ;set addr of env block
      08 AC DD 0076 145 pushl 8(ap) ;set addr of close environment
      52 DD 0079 146 pushl r2 ;set addr of fcb
      08 AE 9F 007B 147 pushab 8(sp) ;set addr of addr
      08 AE 9F 007E 148 pushab 8(sp) ;set addr of addr
      08 AE 9F 0081 149 pushab 8(sp) ;set addr of addr
00000000'GF 06 FB 0084 150 calls #6,g^plis$envir ;process environment
      01 50 E8 008B 151 blbs r0,37$ ;if lbs, cont
      04 008E 152 ret ;error already signaled, just return
      008F 153 37$: $close r3 ;close the file
      50 00000000'8F 50 E8 0098 154 blbs r0,60$ ;if error on close
      08 A2 50 DO 009B 155 movl #pli$ rmsf,r0 ;then set rms fab error code
      51 52 DO 00A2 156 40$: movl r0,fcb_l_error(r2) ;put error in fcb
      51 51 DD 00A9 157 50$: movl r2,r1 ;set fcb address
      50 DD 00AB 158 50$: pushl r1 ;push fcb addr
      00000000'8F 50 DD 00AD 159 pushl r0 ;push error code
      03 6C D1 00B3 161 pushl #pli$error ;set error condition
      06 19 00B6 162 cmpl (ap),#3 ;error level specified?
6E 03 00 0C AC F0 00B8 163 insv 12(ap),#0,#3,(sp) ;set severity to specified level
00000000'GF 03 FB 00BE 164 53$: calls #3,g^plisio_error ;signal the condition
      04 00C5 165 ret ;return
50 00000000'8F 50 DO 00C6 166 55$: movl #pli$_virmemdeal,r0 ;set vir. mem. deallocation error
      D3 11 00CD 167 brb 40$ ;and fail
      00CF 168 ;
      00CF 169 ; deallocate file buffer
      00CF 170 ;
      14 A2 D5 00CF 171 60$: tstl fcb_l_buf(r2) ;if buffer allocated
      27 13 00D2 172 beql 80$ ;then
06 0C A2 0C E0 00D4 173 bbs #atr_v_record,fcb_l_attr(r2),70$ ;if stream
7E 2A A2 3C 00D9 174 movzwl fcb_w_linesize(r2),-(sp) ;set length of buffer
      03 11 00DD 175 brb 75$ ;cont
      18 A2 DD 00DF 176 70$: pushl fcb_l_buf_end(r2) ;push length of buffer
      14 A2 DF 00E2 177 75$: pushal fcb_l_buf(r2) ;push address of buffer
06000000'GF 04 AE DF 00E5 178 pushal 4(sp) ;push address of length
      SE 04 CO 00EF 179 calls #2,g^lib$free_vm ;free buffer
      D1 50 E9 00F2 180 addl2 #4,sp ;
      14 A2 D4 00F5 181 blbc r0,55$ ;if deallocation error then fail
      18 A2 D4 00F8 182 clrl fcb_l_buf(r2) ;indicate buffer not allocated
      0101 C2 94 00FB 183 80$: clrl fcb_l_buf_end(r2) ;
23 0C A2 10 E1 00FF 185 bbc <fcb_5_name+name$esl>(r2) ;clear esl
      38 A2 D5 0104 186 80$: tstl fcb_l_kcb(r2) ;kcb allocated?
      1E 13 0107 187 beql 85$ ;if eql, no, cont
      51 3C A2 9A 0109 188 movzbl fcb_b_numkcbs(r2),r1 ;get number of kcbs
7E 51 2C C5 010D 189 mull3 #kcb_c_len,r1,-(sp) ;get size of kcbs
      38 A2 DF 0111 190 pushal fcb_l_kcb(r2) ;push address of kcbs
      04 AE DF 0114 191 pushal 4(sp) ;push address of length
0000'00'GF 02 FB 0117 192 calls #2,g^lib$free_vm ;free kcbs
      SE 04 CO 011E 193 addl2 #4,sp ;
      A2 50 E9 0121 194 blbc r0,55$ ;if error, fail
      38 A2 D4 0124 195 clrl fcb_l_kcb(r2) ;indicate no kcbs
      0127 196 ;
      0127 197 ; remove file from open file queue, set it not open and return

```

```
52 62 OF 0127 198 :  
50 01 DO 0127 199 85$: remque (r2),2 ;remove fcb from queue  
04 04 012A 200 110$: movl #1,r0 ;set success  
012D 201 ret ;return  
012E 202  
012E 203 .end
```

PLISCLOSE
Symbol table

- pl1 runtime close file

M 14

16-SEP-1984 02:11:51 VAX/VMS Macro V04-00
6-SEP-1984 11:36:29 [PLIRTL.SRC]PLISCLOSE.MAR;1

Page 6
(1)

PL
1-

```

$$TMP1      = 00000001
$$TMP2      = 000'0053
ATR_M_BFALL = 00020000
ATR_M_INPUT = 00000040
ATR_M_OPENED = 00000002
ATR_M_RECORD = 00001000
ATR_V_INDEXED = 00000010
ATR_V_OPENED = 00000001
ATR_V_PRINT = 00000007
ATR_V_RECORD = 00000000
ENV_K_LEN   = 00000022
ENV_L_FILEIDTO = 00000014
ENV_L_FXCTLTO = 00000'8
ENV_L_STATUS = 00000000
ENV_M_CLOSE = 00000040
ENV_Q_CREDATE = 00000004
ENV_Q_EXPDATE = 00000000
ENV_W_OWNGROUP = 0000001E
ENV_W_OWNMEM = 00000020
ENV_W_PROT = 00000010
FCB_B_ENVIR = 00000102
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 = 00000030
FCB_B_RAB = 00000062
FCB_C_LEN = 00000102
FCB_C_STRLN = 00000034
FCB_L_ATTR = 00000000
FCB_L_BUF = 00000014
FCB_L_BUF_END = 00000018
FCB_L_BUF_PT = 00000010
FCB_L_CNDADDR = 000001B2
FCB_L_CONDIT = 000001AE
FCB_L_DTR = 00000010
FCB_L_ERROR = 00000003
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 = 00000020
FCB_W_REVISION = 00000028
KCB_C_LEN = 00000020
KCB_L_DTYP = 00000000
KCB_L_PREC = 00000004
KCB_L_ZERO = 00000028
KCB_W_LEN0 = 0000000A
KCB_W_LEN1 = 0000000E

```

```

KCB_W_LEN2 = 00000012
KCB_W_LEN3 = 00000016
KCB_W_LEN4 = 0000001A
KCB_W_LEN5 = 0000001E
KCB_W_LEN6 = 00000022
KCB_W_LEN7 = 00000026
KCB_W_POS0 = 00000008
KCB_W_POS1 = 00000000
KCB_W_POS2 = 00000010
KCB_W_POS3 = 00000014
KCB_W_POS4 = 00000018
KCB_W_POS5 = 00000010
KCB_W_POS6 = 00000020
KCB_W_POS7 = 00000024
LIB$FREE_VM = ***** X 02
NAM$B_ESC = 00000008
PLI$ENVIR = ***** X 02
PLI$PUT_REC = ***** X 02
PLISCLOSE = 00000000 RG 02
PLI$IO_ERROR = ***** X 02
PLI$ERROR = ***** X 02
PLI$PARM = ***** X 02
PLI$RMSF = ***** X 02
PLI$VIRMEMDEAL = ***** X 02
SIZ... = 00000001
SYSS$CLOSE = ***** GX 02

```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEYE NORD NOWRT NOVEC BYTE
\$AB\$\$	000001C2 (450.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_PLISCODE	0000012E (302.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	9	00:00:00.03	00:00:01.09
Command processing	72	00:00:00.47	00:00:06.16
Pass 1	176	00:00:06.10	00:00:23.63
Symbol table sort	0	00:00:00.63	00:00:02.03
Pass 2	41	00:00:01.05	00:00:04.54
Symbol table output	9	00:00:00.08	00:00:00.08
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	309	00:00:08.38	00:00:37.55

The working set limit was 900 pages.
 30745 bytes (61 pages) of virtual memory were used to buffer the intermediate code.
 There were 30 pages of symbol table space allocated to hold 516 non-local and 16 local symbols.
 203 source lines were read in Pass 1, producing 13 object records in Pass 2.
 18 pages of virtual memory were used to define 16 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1	5
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	13

593 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=TRACEBACK/LIS=LIS\$:PLICLOSE/OBJ=OBJ\$:PLICLOSE MSRC\$:PLICLOSE/UPDATE=(ENH\$:PLICLOSE)+LIB\$:PLIRTMAC/L

