

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

```

```

PPPPPPPPPP
PPPPPPPPPP
PPPPPPPPPP
PPP      PPP
PPP      PPP
PPP      PPP
PPP      PPP
PPP      PPP
PPP      PPP
PPPPPPPPPP
PPPPPPPPPP
PPPPPPPPPP
PPP
PPP
PPP
PPP
PPP
PPP
PPP
PPP
PPP
PPP

```

```

LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL

```

```

IIIIIIIII
IIIIIIIII
IIIIIIIII
III
III
III
III
III
III
III
III
III
III
III
III
III
III
III
III
III
III
III
III
III

```

```

RRRRRRRRRR
RRRRRRRRRR
RRRRRRRRRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRRRRRRRRR
RRRRRRRRRR
RRRRRRRRRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR
RRR      RRR

```

```

TTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT

```

```

LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL

```

```

PPPPPPP LL      IIIIII DDDDDDD EEEEEEEEE LL      EEEEEEEEE TTTTTTTTT EEEEEEEEE
PPPPPPP LL      IIIIII DDDDDDD EEEEEEEEE LL      EEEEEEEEE TTTTTTTTT EEEEEEEEE
PP      PP     LL      II      DD      DD     EE      LL      EE      TT      EE
PP      PP     LL      II      DD      DD     EE      LL      EE      TT      EE
PP      PP     LL      II      DD      DD     EE      LL      EE      TT      EE
PP      PP     LL      II      DD      DD     EE      LL      EE      TT      EE
PPPPPPP LL      II      DD      DD     EEEEEEE LL      EEEEEEE TT      EEEEEEE
PPPPPPP LL      II      DD      DD     EEEEEEE LL      EEEEEEE TT      EEEEEEE
PP      LL      II      DD      DD     EE      LL      EE      TT      EE
PP      LL      II      DD      DD     EE      LL      EE      TT      EE
PP      LL      II      DD      DD     EE      LL      EE      TT      EE
PP      LL      II      DD      DD     EE      LL      EE      TT      EE
PP      LL      II      DD      DD     EE      LL      EE      TT      EE
PP      LLLLLLLLL IIIIII DDDDDDD EEEEEEEEE LLLLLLLLL EEEEEEEEE TT      EEEEEEEEE
PP      LLLLLLLLL IIIIII DDDDDDD EEEEEEEEE LLLLLLLLL EEEEEEEEE TT      EEEEEEEEE

```

```

LL      IIIIII SSSSSSS
LL      IIIIII SSSSSSS
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
LLLLLLLLL IIIIII SSSSSSS
LLLLLLLLL IIIIII SSSSSSS

```

```

0000 1      .title pli$delete - pl1 runtime delete file
0000 2      .ident /1-002/                               ; Edit WHM1002
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :*  ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :*  TRANSFERRED.
0000 17 :*
0000 18 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :*  CORPORATION.
0000 21 :*
0000 22 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27
0000 28
0000 29
0000 30 :++
0000 31 : facility:
0000 32
0000 33 :       VAX/VMS PL1 runtime library.
0000 34
0000 35 : abstract:
0000 36
0000 37 :       This module contains the pl1 runtime routine for deleting a record
0000 38 :       from a file.
0000 39
0000 40 : author: c. spitz 25-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 :--
0000 50
0000 51
0000 52 :+
0000 53 : external definitions
0000 54 :-
0000 55
0000 56 :       $deffcb           ;define file control block offsets
0000 57 :       $rabdef           ;define rab offsets

```

```
0000 58      $rmsdef      ;define rms error codes
0000 59      $fabdef     ;define fab offsets
0000 60
0000 61      ;+
0000 62      ; local definitions
0000 63      ;-
0000 64      $offset 4,positive,<- ;define arguments
0000 65      <fcbaddr,4>,-        ;addr of fcb
0000 66      <keyaddr,4>,-       ;addr of key or keyto
0000 67      <keylen,4>,-       ;length of key or keyto
0000 68      <keytyp,4>,-       ;data type of key or keyto
0000 69      <fastdel,4>,-      ;addr of fast delete
0000 70      <keynum,4>,-       ;addr of key number
0000 71      <recidfrom,4>,-    ;addr of record id from
0000 72      <matchgtr,4>,-    ;addr of match greater
0000 73      <matchgeq,4>,-    ;addr of match greater equal
0000 74      >
0000 75
0000 76      rtshare        ;sharable
0000 77
```

```

0000 79
0000 80 :++
0000 81 : pli$delete -- delete a record from a file
0000 82 :
0000 83 : functional description:
0000 84 :
0000 85 : This routine deletes a record from a pl1 file.
0000 86 :
0000 87 : inputs:
0000 88 : (ap) - number of arguments
0000 89 :         1 if no key and no options
0000 90 :         4 if key and no options
0000 91 :         9 if any options
0000 92 : 4(ap) - addr of fcb
0000 93 : 8(ap) - addr of key
0000 94 : 12(ap) - size/prec of key
0000 95 : 16(ap) - data type of key
0000 96 : 20(ap) - addr of fast delete (aligned bit)
0000 97 : 24(ap) - addr of key number
0000 98 : 28(ap) - addr of record id from
0000 99 : 32(ap) - addr of match greater
0000 100 : 36(ap) - addr of match greater equal
0000 101 :
0000 102 : outputs:
0000 103 : fcb_l_attr
0000 104 : atr_m_write, atr_m_bfall and atr_m_virgin are set false
0000 105 : atr_m_delete and atr_m_currec are set true
0000 106 : fcb_q_rfa is set to the rfa of the record that was deleted
0000 107 :
0000 108 : side effects:
0000 109 : the record is deleted from the file.
0000 110 : --
0000 111 :
007C 0000 112 : .entry pli$delete,^m<r2,r3,r4,r5,r6>
0002 113 :
0002 114 : check arguments. there must be at least one argument.
0002 115 :
0002 116 :     cmpl    (ap),#1                ;enough arguments?
0005 117 :     bgeq   10$                    ;geq, yes
0007 118 :     crrl   r0                      ;indicate not enough args
0009 119 :     brw    fail                    ;and fail
000C 120 10$:   movl   fcbaddr(ap),r2      ;get address of fcb
0010 121 :     movl   fcb_l_attr(r2),r3       ;get attributes
0014 122 :
0014 123 : open the file if necessary. the file will be opened with record and
0014 124 : update. if the file is not opened after calling open, an error is
0014 125 : signaled.
0014 126 :
0014 127 :     bbs    #atr_v_opened,r3,20$    ;if file not opened
0018 128 :     pushl  #<atr_m_record!atr_m_update> ;set record and update for implied
001E 129 :     ;open
001E 130 :     pushl  r2                      ;push address of fcb
0020 131 :     calls  #2,q^pli$open           ;open the file
0027 132 :     movl   fcb_l_attr(r2),r3       ;get attributes
002B 133 :     bbs    #atr_v_opened,r3,20$    ;if file still not opened
002F 134 :     movl   #pli$_open,r0          ;set open failure
0124 31 0036 135 :     brw    fail                    ;and fail

```

```

0039 136 :
0039 137 : make sure file has proper attributes. file must have record and
0039 138 : update. file must not have delete.
0039 139 : if key is specified, file must have keyed.
0039 140 :
50 0A 53 0C E0 0039 141 20$: bbs #atr_v_record,r3,30$ ;if file doesn't have record
00000000'8F DO 003D 142 movl #pli$_notrec,r0 ;set not record file
0116 31 0044 143 brw fail ;and fail
0A 53 04 E0 0047 144 30$: bbs #atr_v_update,r3,40$ ;if file doesn't have update
00000000'8F DO 004B 145 movl #pli$_notupdate,r0 ;set not update file
0108 31 0052 146 brw fail ;and fail
0A 00BC C2 05 E1 0055 147 40$: bbc #fab$v_bio,<fcb_b_fab+fab$b_fac>(r2),45$ ;if file open for
00000000'8F DO 005B 148 ;block io
00F8 31 0062 149 movl #pli$_conblockio,r0 ;set conflicts with block io
0065 150 brw fail ;and fail
0065 151 :
0065 152 : process options
0065 153 :
04 A4 54 62 A2 9E 0065 154 45$: movab fcb_b_rab(r2),r4 ;get address of rab
00600000 8F CA 0069 155 bicl #<rab$m_kge!rab$m_kgt>,rab$l_rop(r4) ;clear kge and kgt
04 04 6C D1 0071 156 cmpl (ap),#4 ;options specified?
03 18 0074 157 bgeq 50$ ;if geq, maybe
0094 31 0076 158 brw 160$ ;its lss, so process as seql
09 63 13 0079 159 50$: beql 120$ ;if eql, only key specified, go do it
00000000'8F DO 007B 160 cmpl (ap),#9 ;enuf options?
00A 13 007E 161 beql 60$ ;if eql, then yes
00D3 31 0080 162 movl #pli$_invnumopt,r0 ;set invalid options
0087 163 brw fail ;and fail
008A 164 :
008A 165 : process match greater and match greater equal
008A 166 :
50 20 AC DO 008A 167 60$: movl matchgtr(ap),r0 ;get addr of match greater
00000000'GF 16 008E 168 jsb g^pli$$matchgtr ;process match greater
50 24 AC DO 0094 169 movl matchgeq(ap),r0 ;get addr of match greater equal
00000000'GF 16 0098 170 jsb g^pli$$matchgeq ;process match greater equal
009E 171 :
009E 172 : process key number
009E 173 :
50 18 AC DO 009E 174 movl keynum(ap),r0 ;get addr of key num
51 08 AC DO 00A2 175 movl keyaddr(ap),r1 ;get addr of key
00C00000'GF 16 00A6 176 jsb g^pli$$keynum ;proces key number
00AC 177 :
00AC 178 : process fast delete
00AC 179 :
04 A4 00000040 8F CA 00AC 180 bicl #rab$m_fdl,rab$l_rop(r4) ;assume no options
50 14 AC DO 00B4 181 movl fastdel(ap),r0 ;get addr of fast del
14 13 00B8 182 beql 100$ ;if eql, not specified
0A 53 10 E0 00BA 183 bbs #atr_v_indexed,r3,90$ ;if file doesn't have indexed
50 00000000'8F DO 00BE 184 movl #pli$_notindexed,r0 ;set not indexed
0095 31 00C5 185 brw fail ;and fail
04 A4 01 06 60 F0 00C8 186 90$: insv (r0),#rab$v_fdl,#1,rab$l_rop(r4) ;set fast delete in rab
00CE 187 :
00CE 188 : process record id from
00CE 189 :
50 1C AC DO 00CE 190 100$: movl recidfrom(ap),r0 ;get rfa from addr
51 08 AC DO 00D2 191 movl keyaddr(ap),r1 ;get addr of key
00000000'GF 16 00D6 192 jsb g^pli$$recidfrom ;process rfa from

```

```

      4E 12 00DC 193      bneq 180$      ;if neq, do rfa find
08 AC D5 00DE 194 120$: tstl keyaddr(ap) ;key specified?
      2A 13 00E1 195      beql 160$      ;if eql, no, do seql delete
      00E3 196      ;
      00E3 197      ; process key option.
      00E3 198      ;
50 0A 53 08 E0 00E3 199 130$: bbs #atr_v_keyed,r3,140$ ;if file doesn't have keyed
00000000'8F D0 00E7 200      ;set not keyed file
      006C 31 00EE 201      ;and fail
50 08 AC 9E 00F1 202 140$: movab keyaddr(ap),r0 ;point to key descr
00000000'GF 16 00F5 203      jsb g^pli$$readkey,r6 ;process key
00000000'GF 16 00FB 204      jsb g^pli$$smallgef ;find the record
      35 11 0101 205      brb 190$ ;cont
50 00000000'8F D0 0103 206 150$: movl #pli$_rmsr,r0 ;set rms rab error
      0050 31 010A 207      brw fail ;and fail
      010D 208      ;
      010D 209      ; for sequential delete, find current record if necessary.
      010D 210      ;
50 0A 53 13 E1 010D 211 160$: bbc #atr_v_delete,r3,170$ ;if file has delete
00000000'8F D0 0111 212 165$: movl #pli$_nocurrec,r0 ;set no current record
      0042 31 0118 213      brw fail ;and fail
19 53 12 E1 011B 214 170$: bbc #atr_v_currec,r3,190$ ;if current record not correct
1E A4 02 90 011F 215      movb #rab$c_rfa,rab$b,rac(r4) ;set rfa access in rab
10 A4 20 A2 7D 0123 216      movq fcb_q_rfa(r2),rab$w_rfa(r4) ;set rfa in rab
E5 53 19 E0 0128 217      bbs #atr_v_virgin,r3,165$ ;if open just occurred, fail
      CB 50 E9 0135 218 180$: $find r4 ;find the current record
      0138 219      blbc r0,150$ ;if lbc, fail
      0138 220      ;
      0138 221      ; delete the record
      0138 222      ;
20 A2 10 A4 7D 0138 223 190$: movq rab$w_rfa(r4),fcb_q_rfa(r2) ;copy correct current record's rfa
      013D 224      ;to fcb
      013D 225      $delete r4 ;delete the record
      BA 50 E9 0146 226      blbc r0,150$ ;if lbc, fail
OC A2 02120000 8F CA 0149 227      bicl #<atr_m_write!atr_m_bfall! - ;'deallocate buffer', clear write
      000C0000 8F C8 0151 228      atr_m_virgin>,fcb_l_attr(r2) ;and virgin
      OC A2 C8 0151 229      bisl #<atr_m_delete!atr_m_currec>,- ;set current record incorrect
      50 01 D0 0159 230      fcb_l_attr(r2) ;and delete and copy atr to fcb
      04 015C 231 200$: movl #1,r0 ;set success
      015D 232      ret ;return
      015D 233
```

```

50 00000000'0B 12 015D 235
      8F D0 015D 236 fail: bneq 10$ ;if not enough parms
      52 D4 015F 237 movl #pli$_parm,r0 ;set not enough parms
      1C 11 0166 238 clrl r2 ;set no fcb
      08 A2 50 D0 016A 239 brb 40$ ;just signal error
50 00000000'8F D1 016E 240 10$: movl r0,fcbl_error(r2) ;set error code in fcb
      OF D1 016E 241 cmpl #pli$_rmsr,r0 ;rms rab error code?
      01 6C D1 0175 242 bneq 40$ ;if neq, no, cont
      53 08 AC 9E 017A 243 cmpl (ap),#1 ;key passed?
      00000000'GF 16 017C 244 beql 40$ ;if eql, no, cont
      52 DD 0180 245 movab keyaddr(ap),r3 ;set key address for onkey
      50 DD 0186 246 jsb g^pli$$chk_keycnd ;check for key condition
00000000'8F 03 FB 0188 247 40$: pushl r2 ;set fcb addr
      04 04 018A 248 pushl r0 ;set error code
      0197 249 50$: pushl #pli$_error ;set error condition
      0198 250 calls #3,g^pli$io_error ;signal the condition
      0198 251 ret ;return
      0198 252
      0198 253 .end
```



PLI\$DELETE  
Symbol table

- pl1 runtime delete file

L 14

16-SEP-1984 02:17:20 VAX/VMS Macro V04-00  
6-SEP-1984 11:37:31 [PLIRTL.SRC]PLI\$DELETE.MAR;1

Page 7  
(1)

```

SS.TMP1           = 00000001
SS.TMP2           = 00000054
ATR_M_BFALL      = 00020000
ATR_M_CURREC     = 00040000
ATR_M_DELETE     = 00080000
ATR_M_RECORD     = 00001000
ATR_M_UPDATE     = 00000010
ATR_M_VIRGIN     = 02000000
ATR_M_WRITE      = 00100000
ATR_V_CURREC     = 00C00012
ATR_V_DELETE     = 00000013
ATR_V_INDEXED    = 00000010
ATR_V_KEYED      = 00000008
ATR_V_OPENED     = 00000001
ATR_V_RECORD     = 0000000C
ATR_V_UPDATE     = 00000004
ATR_V_VIRGIN     = 00000019
DIR...           = 00000001
FABS$B_FAC       = 00000016
FABS$V_BIO       = 00000005
FAIL             = 0000015D R    0?
FASTDEL          = 00000014
FCBADDR          = 00000004
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_STRLLEN   = 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_PRI        = 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
KEYADDR          = 00000008
KEYLEN           = 0000000C
KEYNUM           = 00000018
    
```

```

KEYTYP           00000010
MATCHGEQ         00000024
MATCHGTR         00000020
PLI$$CHK_KEYCND ***** X 02
PLI$$KEYRU...    ***** X 02
PLI$$MATCHGEQ    ***** X 02
PLI$$MATCHGTR    ***** X 02
PLI$$READKEY_R6 ***** X 02
PLI$$RECIDFROM   ***** X 02
PLI$$SMALLGET    ***** X 02
PLI$DELETE       00000000 RG
PLI$IO_ERROR     ***** X 02
PLI$OPEN         ***** X 02
PLI$_CONBLOKIO  ***** X 02
PLI$_ERROR       ***** X 02
PLI$_INVNUMOPT   ***** X 02
PLI$_NOCURREC   ***** X 02
PLI$_NOTINDEXED ***** X 02
PLI$_NOTKEYD     ***** X 02
PLI$_NOTREC      ***** X 02
PLI$_NOTUPDATE   ***** X 02
PLI$_OPEN        ***** X 02
PLI$_PARM        ***** X 02
PLI$_RMSR        ***** X 02
RABS$RAC         = 0000001E
RABS$RFA         = 00000002
RABS$L_RFP       = 00000004
RABS$M_FDL       = 0000004C
RABS$M_KGE       = 00200000
RABS$M_KGT       = 00400000
RABS$V_FDL       = 00000006
RABS$W_RFA       = 00000010
RECIDFROM        = 0000001C
SI7...           = 00000001
SYS$DELETE       ***** GX 02
SYS$FIND         ***** GX 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
\$ABSS	000001C2 ( 450.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_PLI\$CODE	00000198 ( 408.)	02 ( 2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	9	00:00:00.06	00:00:01.21
Command processing	67	00:00:00.55	00:00:03.91
Pass 1	190	00:00:07.03	00:00:20.48
Symbol table sort	0	00:00:00.76	00:00:01.93
Pass 2	49	00:00:01.32	00:00:05.48
Symbol table output	11	00:00:00.08	00:00:00.09
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	328	00:00:09.84	00:00:33.16

The working set limit was 1050 pages.  
37813 bytes (74 pages) of virtual memory were used to buffer the intermediate code.  
There were 40 pages of symbol table space allocated to hold 669 non-local and 22 local symbols.  
253 source lines were read in Pass 1, producing 14 object records in Pass 2.  
19 pages of virtual memory were used to define 17 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1	4
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	14

760 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=TRACEBACK/LIS=LIS\$:PLI\$DELETE/OBJ=OBJ\$:PLI\$DELETE MSRC\$:PLI\$DELETE/UPDATE=(ENH\$:PLI\$DELETE)+LIB\$:PLIRTM



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

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 144 terminal windows arranged in 12 rows and 12 columns. Each window contains text-based data or program output. Several windows are clearly labeled with titles:

- Row 4, Column 2: **PLICONTROL LIS**
- Row 5, Column 10: **PLIDELETE LIS**
- Row 5, Column 11: **PLIDATA LIS**
- Row 7, Column 10: **PLIDATE LIS**
- Row 8, Column 8: **PLICUTPIC LIS**
- Row 9, Column 11: **PLIENUR LIS**

The remaining windows contain various data outputs, including lists of numbers, text blocks, and program headers, though they are less legible due to the low resolution and high density of the grid.