


```

PPPPPPPP          LL          IIIIII          PPPPPPPP          UU          UU          TTTTTTTTTTTT          LL          IIIIII          SSSSSSSS
PPPPPPPP          LL          IIIIII          PPPPPPPP          UU          UU          TTTTTTTTTTTT          LL          IIIIII          SSSSSSSS
PP          PP      LL          II          PP          PP      UU          UU          TT          LL          II          SS
PP          PP      LL          II          PP          PP      UU          UU          TT          LL          II          SS
PP          PP      LL          II          PP          PP      UU          UU          TT          LL          II          SS
PP          PP      LL          II          PPPPPPPP          UU          UU          TT          LL          II          SSSSSS
PPPPPPPP          LL          IIIIII          PPPPPPPP          UU          UU          TT          LL          II          SSSSSS
PP          LL          II          PP          UU          UU          TT          LL          II          SS
PP          LL          II          PP          UU          UU          TT          LL          II          SS
PP          LL          II          PP          UU          UU          TT          LL          II          SS
PP          LL          IIIIII          PP          UU          UU          TT          LL          IIIIII          SSSSSSSS
PP          LL          IIIIII          PP          UUUUUUUUUU          TT          LL          IIIIII          SSSSSSSS
PP          LLLLLLLLLL          IIIIII          PP          UUUUUUUUUU          TT          LLLLLLLLLL          IIIIII          SSSSSSSS
PP          LLLLLLLLLL          IIIIII          PP          UUUUUUUUUU          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          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS

```

```

0000 1      .title pli$putlistitem
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 : 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 put items to a
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      $rabdef        ;define rms rab offsets
0000 60      $rmsdef        ;define rms error codes
0000 61      $sssdef        ;define system status codes
0000 62
0000 63
0000 64      ; local data
0000 65      ;
0000 66
0000 67      rtshare          ;sharable
0000 68
0000 69      ;++
0000 70      ; pli$putl****
0000 71
0000 72      ; the pli$putl**** routines are called by the compiled code to put items
0000 73      ; to a stream output file under list directed transmission. each routine
0000 74      ; converts the source item to a character string based on the source data
0000 75      ; type, and puts then puts the string to the file by jumping to
0000 76      ; pli$$putnlis_r6.
0000 77      ;--
0000 78
0000 79
0000 80      ;pli$putlchar_r6
0000 81      ; inputs:
0000 82      ;     r0 - address of element to put
0000 83      ;     r1 - size/prec of element to put
0000 84      ;     r11 - address of stream block
0000 85      ;     ap - address of file control block
0000 86      ; outputs:
0000 87      ;     none
0000 88      ; side effects:
0000 89      ;     r0-r6 are destroyed
0000 90
0000 91      pli$putlchar_r6::
0000 92      bisl      #atr_m_recur,fcbl_attr(ap) ;set recursion flag
0000 93      addl      r0,r1 ;get ending address of source
0000 94      movab     <str_b_field+2>(r11),r2 ;get starting addr in field
0000 95      movl      r2,r4 ;copy it
0000 96      movl      str_l_fld_end(r11),r3 ;get end addr of field
0000 97      bbs      #atr_v_print,fcbl_attr(ap),5$ ;skip lead quote if print
0000 98      movb      #^x27,(r2)+ ;insert the leading quote
0000 99      5$:      cmpl      r0,r1 ;nothing in source?
0000 100     beql      60$ ;if eql, then yes
0000 101     10$:     cmpb      (r0),#^x27 ;next char a quote?
0000 102     bneq     20$ ;if neg, then no
0000 103     bbs      #atr_v_print,fcbl_attr(ap),20$ ;if print file, don't change
0000 104     movw     #^x2727,(r2)+ ;insert 2 quotes
0000 105     brb      50$ ;continue
0000 106     20$:     movb      (r0),(r2)+ ;copy to field
0000 107     50$:     aoblss    r1,r0,55$ ;if not end of source, cont
0000 108     brb      60$ ;finish it off
0000 109     55$:     cmpl      r2,r3 ;field overflow?
0000 110     blss     10$ ;if lss, then no, cont
0000 111     movl     #pli$_strovfl,r0 ;set field overflow
0000 112     brw      fail ;and fail
0000 113     60$:     bbs      #atr_v_print,fcbl_attr(ap),70$ ;if print, don't add trail quote
0000 114     movb      #^x27,(r2)+ ;insert trailing quote

```

```

0C AC 08 C8
51 50 C0
52 1A AB 9E
54 52 D0
53 14 AB D0
03 0C AC 07 E0
82 27 90
51 50 D1
27 29 13
27 60 91
07 0C AC 07 E0
82 2727 8F B0
82 03 11
02 50 51 F2
53 0F 11
50 00000000 8F D0
0113 31
03 0C AC 07 E0
82 27 90

```

```

      52 54 C2 0050 115 70$: subl r4,r2 ;get length
      74 52 B0 0053 116      movw r2,-(r4) ;set length in field
00000000'GF 16 U056 117      jsb g^pli$$putnlis_r6 ;put in buffer
      OC AC 08 CA 005C 118      bicl #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
      05 0060 119      rsb ;return
      0061 120
      0061 121 ;pli$putlvcha_r6
      0061 122 :inputs:
      0061 123 : r0 - address of element to put
      0061 124 : r1 - size/prec of element to put
      0061 125 : r11 - address of stream block
      0061 126 : ap - address of file control block
      0061 127 :outputs:
      0061 128 : none
      0061 129 :side effects:
      0061 130 : r0-r6 are destroyed
      0061 131 :
      0061 132
      0061 133 pli$putlvcha_r6::
      OC AC 08 C8 0061 134      bisl #atr_m_recur,fcbl_attr(ap) ;set recursion flag
      51 80 3C 0065 135      movzwl (r0)+,r1 ;get length of source
      FF95 31 0068 136      brw pli$putlchar_r6 ;continue in common
      006B 137
      006B 138
      006B 139
      006B 140 ;pli$putlbit_r6
      006B 141 :inputs:
      006B 142 : r0 - address of element to put
      006B 143 : r1 - size/prec of element to put
      006B 144 : r2 - offset to starting bit
      006B 145 : r11 - address of stream block
      006B 146 : ap - address of file control block
      006B 147 :outputs:
      006B 148 : none
      006B 149 :side effects:
      006B 150 : r0-r6 are destroyed
      006B 151 :
      006B 152
      006B 153 pli$putlbit_r6::
      OC AC 08 C8 006B 154      bisl #atr_m_recur,fcbl_attr(ap) ;set recursion flag
      55 52 D0 006F 155      movl r2,r5 ;copy offset
      52 18 AB 9E 0072 156      movab str_b_field(r11),r2 ;get field addr
      82 03 51 A1 0076 157      addw3 r1,#3,(r2)+ ;set size
      82 27 90 007A 158      movb #^x27,(r2)+ ;insert a quote
      53 51 D0 007D 159      movl r1,r3 ;get size
      000003E8 8F 53 D1 0080 160      cmpl r3,#1000 ;field overflow?
      50 00000000'8F 0A 15 0087 161      bleq 10$ ;if leg, then no
      54 52 51 C1 0093 164 10$: addl3 r1,r2,r4 ;get addr of end of string
      64 4227 8F B0 0097 165      movw #^x4227,(r4) ;plug in trailing quote and B
      00000000'GF 00 FB 009C 166      calls #0,g^pli$bitchar_r6 ;convert bits
      00000000'GF 16 00A3 167      jsb g^pli$$putnlis_r6 ;put in buffer
      OC AC 08 CA 00A9 168      bicl #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
      05 00AD 169      rsb ;return
      00AE 170
      00AE 171
```

```
00AE 172 ;pli$putlbit_r6
00AE 173 :   inputs:
00AE 174 :   r0 - address of element to put
00AE 175 :   r1 - size/prec of element to put
00AE 176 :   r11 - address of stream block
00AE 177 :   ap - address of file control block
00AE 178 :   outputs:
00AE 179 :   none
00AE 180 :   side effects:
00AE 181 :   r0-r6 are destroyed
00AE 182 :
00AE 183 :
00AE 184 pli$putlbit_r6::
OC AC 08 C8 00AE 185      bisl   #atr_m_recur,fcbl_attr(ap) ;set recursion flag
      52 D4 00B2 186      clrl   r2           ;set offset to 0
      B5 11 00B4 187      brb    pli$putlbit_r6      ;join common code
00B6 188
00B6 189 ;pli$putlfixb_r6
00B6 190 :   inputs:
00B6 191 :   r0 - address of element to put
00B6 192 :   r1 - size/prec of element to put
00B6 193 :   r11 - address of stream block
00B6 194 :   ap - address of file control block
00B6 195 :   outputs:
00B6 196 :   none
00B6 197 :   side effects:
00B6 198 :   r0-r6 are destroyed
00B6 199 :
00B6 200 :
00B6 201 pli$putlfixb_r6::
      OC AC 08 C8 00B6 202      bisl   #atr_m_recur,fcbl_attr(ap) ;set recursion flag
      52 18 AB 9E 00BA 203      movab  str_b_field(r11),r2      ;set field addr
53 000003EB 8F D0 00BE 204      movl   #1000,r3           ;set size
00000000'GF 00 FB 00C5 205      calls  #0,g^pli$fixbvcha_r6      ;convert it
      00000000'GF 16 00CC 206      jsb   g^pli$putnlis_r6      ;put in buffer
      OC AC 08 CA 00D2 207      bicl   #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
      05 00D6 208      rsb    ;return
00D7 209
00D7 210
00D7 211 ;pli$putlfixd_r6
00D7 212 :   inputs:
00D7 213 :   r0 - address of element to put
00D7 214 :   r1 - size/prec of element to put
00D7 215 :   r11 - address of stream block
00D7 216 :   ap - address of file control block
00D7 217 :   outputs:
00D7 218 :   none
00D7 219 :   side effects:
00D7 220 :   r0-r6 are destroyed
00D7 221 :
00D7 222 :
00D7 223 pli$putlfixd_r6::
      OC AC 08 C8 00D7 224      bisl   #atr_m_recur,fcbl_attr(ap) ;set recursion flag
      52 18 AB 9E 00DB 225      movab  str_b_field(r11),r2      ;set field addr
53 000003EB 8F D0 00DF 226      movl   #1000,r3           ;set size
00000000'GF 00 FB 00E6 227      calls  #0,g^pli$fixdvcha_r6      ;convert it
      00C00000'GF 16 00ED 228      jsb   g^pli$putnlis_r6      ;put in buffer
```

```
OC AC 08 CA 00F3 229      bicl  #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
05 00F7 230      rsb          ;return
00F8 231
00F8 232
00F8 233      :pli$putlfltb_r6
00F8 234      :inputs:
00F8 235      r0 - address of element to put
00F8 236      r1 - size/prec of element to put
00F8 237      r11 - address of stream block
00F8 238      ap - address of file control block
00F8 239      :outputs:
00F8 240      none
00F8 241      :side effects:
00F8 242      r0-r6 are destroyed
00F8 243      :
00F8 244
00F8 245      pli$putlfltb_r6::
OC AC 08 C8 00F8 246      bisl  #atr_m_recur,fcbl_attr(ap) ;set recursion flag
53 52 18 AB 9E 00FC 247      movab str_b_field(r11),r2 ;set field addr
00000000'GF 8F D0 0100 248      movl  #1000,r3 ;set field width
00000000'GF 00 FB 0107 249      calls #0,g^pli$fltbvcha_r6 ;convert it
OC AC 08 16 010E 250      jsb   g^pli$$putnlis_r6 ;put in buffer
05 CA 0114 251      bicl  #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
05 0118 252      rsb          ;return
0119 253
0119 254
0119 255      :pli$putlfltd_r6
0119 256      :inputs:
0119 257      r0 - address of element to put
0119 258      r1 - size/prec of element to put
0119 259      r11 - address of stream block
0119 260      ap - address of file control block
0119 261      :outputs:
0119 262      none
0119 263      :side effects:
0119 264      r0-r6 are destroyed
0119 265      :
0119 266
0119 267      pli$putlfltd_r6::
OC AC 08 C8 0119 268      bisl  #atr_m_recur,fcbl_attr(ap) ;set recursion flag
53 52 18 AB 9E 011D 269      movab str_b_field(r11),r2 ;set field addr
00000000'GF 8F D0 0121 270      movl  #1000,r3 ;set field width
00000000'GF 00 FB 0128 271      calls #0,g^pli$fltdvcha_r6 ;convert it
OC AC 08 16 012F 272      jsb   g^pli$$putnlis_r6 ;put in buffer
05 CA 0135 273      bicl  #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
05 0139 274      rsb          ;return
013A 275
013A 276
013A 277      :pli$putlpic_r6
013A 278      :inputs:
013A 279      r0 - address of element to put
013A 280      r1 - size/prec of element to put
013A 281      r11 - address of stream block
013A 282      ap - address of file control block
013A 283      :outputs:
013A 284      none
013A 285      :side effects:
```


PLISPUTLISTITEM
Symbol table

F 13

ATR_M_RECUR	=	00000008		
ATR_V_PRINT	=	00000007		
FAIC		0000015B	R	02
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		000000FE		
FCB_B_NUMKCBS		0000003C		
FCB_B_RAB		00000062		
FCB_C_LEN		000001C2		
FCB_C_STRLIN		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_DTR		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		
PLISSPOTNLIS_R6		*****	X	02
PLISBITCHAR_R6		*****	X	02
PLISFIXBVCHA_R6		*****	X	02
PLISFIXDVCHA_R6		*****	X	02
PLISFLTBVCHA_R6		*****	X	02
PLISFLTDVCHA_R6		*****	X	02
PLISIO_ERROR		*****	X	02
PLISPIVCHA_R6		*****	X	02
PLISPUTLABIT_R6		000000AE	RG	02
PLISPUTLBIT_R6		0000006B	RG	02
PLISPUTLCHAR_R6		00000000	RG	02
PLISPUTLFIXB_R6		000000B6	RG	02
PLISPUTLFIXD_R6		000000D7	RG	02
PLISPUTLFLTBR6		000000F8	RG	02
PLISPUTLFLTD_R6		00000119	RG	02
PLISPUTLPIC_R6		0000013A	RG	02
PLISPUTLVCHA_R6		00000061	RG	02
PLIS_ERROR		*****	X	02

PLIS_STROVFL		*****	X	02
SIZ...	=	00000001		
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		
STR_B_FIELD		00000018		
STR_C_LEN		00000008		
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		00000004		
STR_L_STACK_END		00000408		

PL
Sy
RM
SI
SY
SY
PS
--
SA
-F
Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As
Th
44
Th
43
23
Ma
--
-9
-9
TC
80
Th
MA

! 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	FFFFFFFFC (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_PLISCODE	00000171 (369.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	12	00:00:00.08	00:00:00.29
Command processing	74	00:00:00.62	00:00:01.95
Pass 1	251	00:00:09.47	00:00:19.42
Symbol table sort	0	00:00:01.34	00:00:02.66
Pass 2	59	00:00:01.62	00:00:03.81
Symbol table output	10	00:00:00.08	00:00:00.28
Psect synopsis output	1	00:00:00.03	00:00:00.23
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	407	00:00:13.24	00:00:28.65

The working set limit was 1050 pages.
51576 bytes (101 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 972 non-local and 8 local symbols.
304 source lines were read in Pass 1, producing 11 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	6
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	13

995 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

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

