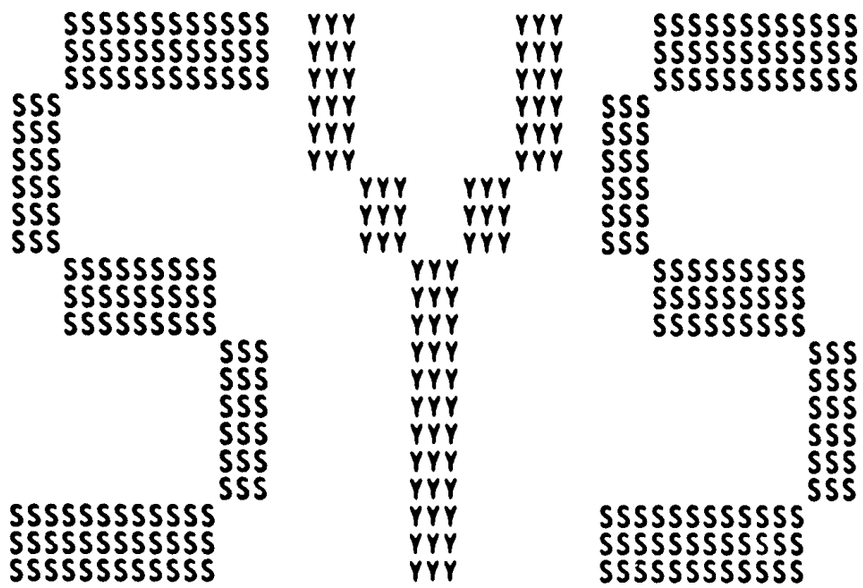

FILEID**CONSOLIO

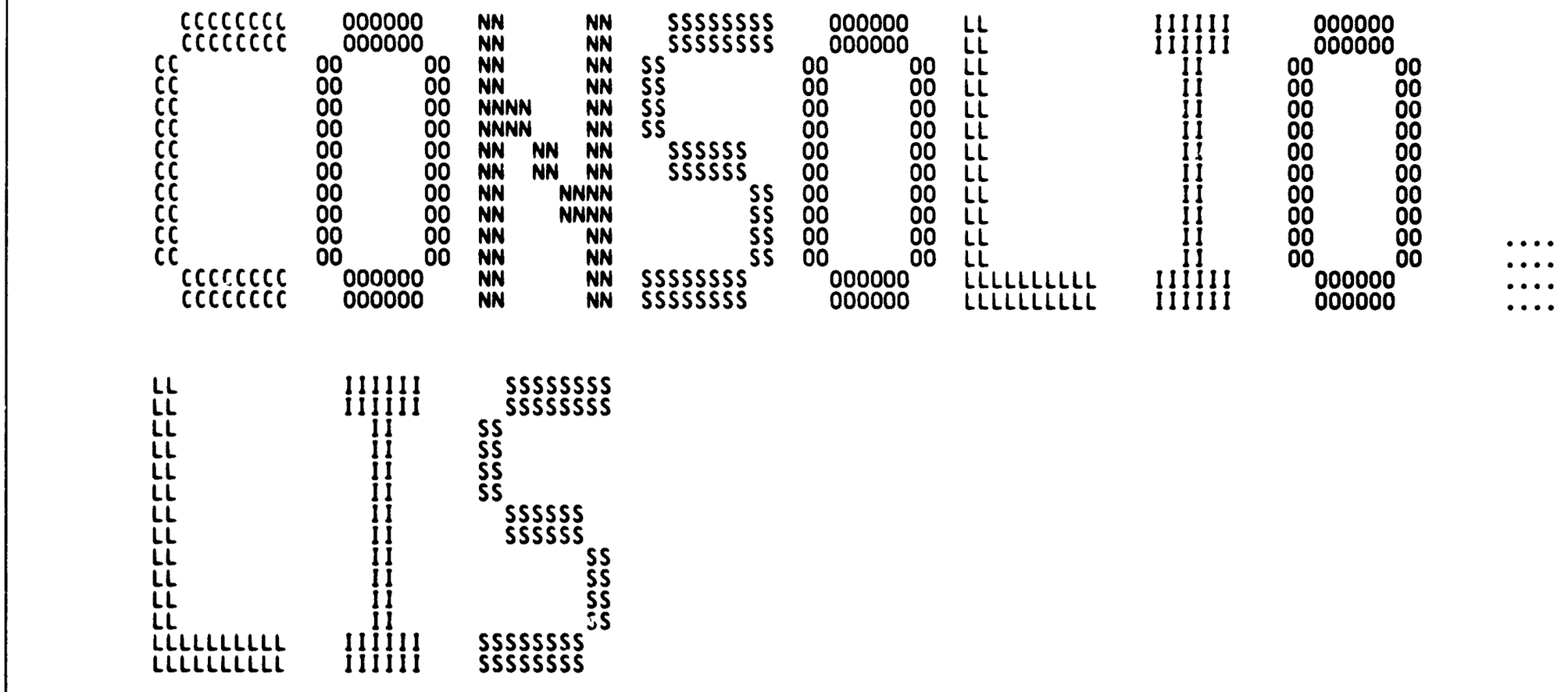

Table of contents

```
                        .TITLE CONSOLIO - CONSOLE TERMINAL I/O ROUTINES
                        .IDENT  'V04-000'

                ;
                ;*****************************************************************************
                ;*                                                                           *
                ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
                ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
                ;*  ALL RIGHTS RESERVED.                                                     *
                ;*                                                                           *
                ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
                ;*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE           *
                ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER      *
                ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
                ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY       *
                ;*  TRANSFERRED.                                                             *
                ;*                                                                           *
                ;*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE     *
                ;*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT         *
                ;*  CORPORATION.                                                             *
                ;*                                                                           *
                ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS      *
                ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
                ;*                                                                           *
                ;*****************************************************************************
                ;
                ; D. N. CUTLER 12-AUG-76
                ;
                ; CONSOLE I/O ROUTINES
                ;
                ; Modified by:
                ;
                ;       V03-001 WHM0001         Bill Matthews           9-Jul-1984
                ;               Add support for alternate console terminal.
                ;
                ;       V02-004 TCM0002         Trudy C. Matthews       10-Aug-1981
                ;               Remember to strip parity bit when checking for XON/XOFF.
                ;
                ;       V02-003 TCM0001         Trudy C. Matthews       14-May-1981
                ;               Added XON/XOFF recognition during terminal I/O.
                ;
                ; MACRO LIBRARY CALLS
                ;

                        $PRDEF                          ;DEFINE PROCESSOR REGISTERS

                ;
                ; LOCAL SYMBOLS
                ;
                ; CHARACTER DEFINITIONS
                ;

00000020        BLANK=32                                ;BLANK
0000000D        CR=13                                   ;CARRIAGE RETURN
0000000A        LF=10                                   ;LINE FEED
```

```
00000037  0000     58 SEVEN=55                                ;DIGIT 7
00000030  0000     59 ZERO=48                                 ;DIGIT 0
00000013  0000     60         CONTROL_S = 19
00000011  0000     61         CONTROL_Q = 17
          0000     62
          0000     63 ;
          0000     64 ; TERMINAL REGISTER DEFINITIONS
          0000     65 ;
          0000     66
FFFFFFFC  0000     67 RCSR=-4                                 ;OFFSET RECEIVER CSR
FFFFFFFE  0000     68 RDBR=-2                                 ;OFFSET RECEIVER DBR
00000000  0000     69 TCSR=0                                  ;OFFSET TRANSMITTER CSR
00000002  0000     70 TDBR=2                                  ;OFFSET TRANSMITTER DBR
          0000     71
          0000     72 ;
          0000     73 ; LOCAL DATA
          0000     74 ;
          0000     75 ; HEX CONVERSION TABLE
          0000     76 ;
          0000     77
        00000000   78          .PSECT  $AEXENONPAGED
          0000     79 EXE$AB_HEXTAB::                         ;HEXDECIMAL CONVERSION TABLE
       30 0000     80          .ASCII  /0/                    ;0 DIGIT
       31 0001     81          .ASCII  /1/                    ;1 DIGIT
       32 0002     82          .ASCII  /2/                    ;2 DIGIT
       33 0003     83          .ASCII  /3/                    ;3 DIGIT
       34 0004     84          .ASCII  /4/                    ;4 DIGIT
       35 0005     85          .ASCII  /5/                    ;5 DIGIT
       36 0006     86          .ASCII  /6/                    ;6 DIGIT
       37 0007     87          .ASCII  /7/                    ;7 DIGIT
       38 0008     88          .ASCII  /8/                    ;8 DIGIT
       39 0009     89          .ASCII  /9/                    ;9 DIGIT
       41 000A     90          .ASCII  /A/                    ;10 DIGIT
       42 000B     91          .ASCII  /B/                    ;11 DIGIT
       43 000C     92          .ASCII  /C/                    ;12 DIGIT
       44 000D     93          .ASCII  /D/                    ;13 DIGIT
       45 000E     94          .ASCII  /E/                    ;14 DIGIT
       46 000F     95          .ASCII  /F/                    ;15 DIGIT
```

```
                     0010    97          .SBTTL  CONVERT LONGWORD TO HEX AND OUTPUT DIGITS
                     0010    98 ;+
                     0010    99 ; EXE$OUTHEX - CONVERT LONGWORD TO HEX AND OUTPUT DIGITS
                     0010   100 ;
                     0010   101 ; THIS ROUTINE IS CALLED VIA A JSB TO CONVERT A LONGWORD TO HEX AND
                     0010   102 ; OUTPUT THE RESULTANT DIGITS TO A SPECIFIED DEVICE IN NONINTERRUPT
                     0010   103 ; MODE.
                     0010   104 ;
                     0010   105 ; INPUTS:
                     0010   106 ;
                     0010   107 ;       R1 = VALUE TO BE CONVERTED.
                     0010   108 ;       R11 = OUTPUT DEVICE CSR ADDRESS.
                     0010   109 ;
                     0010   110 ; OUTPUTS:
                     0010   111 ;
                     0010   112 ;       THE SPECIFED VALUE IS CONVERTED TO HEX AND OUTPUT.
                     0010   113 ;-
                00000000    114          .PSECT  Z$INIT__BUGC            ;
                     0000   115
                     0000   116 EXE$OUTHEX::                             ;CONVERT AND OUTPUT HEX LONGWORD
          52    1C   D0  0000   117          MOVL    #28,R2                 ;SET POSITION OF FIRST FOUR BIT FIELD
    50    51    04   52  EF  0003   118 10$:     EXTZV   R2,#4,R1,R0            ;GET NEXT FOUR BITS OF VALUE
    50  00000000'9F40  9A  0008   119          MOVZBL  @#EXE$AB_HEXTAB[R0],R0 ;GET ASCII CHARACTER
                  0D   10  0010   120          BSBB    EXE$OUTCHAR            ;OUTPUT DIGIT
FFEA 52    FC 8F   00   9D  0012   121          ACBB    #0,#-4,R2,10$          ;ANY MORE DIGITS TO CONVERT?
                     05  0019   122          RSB                            ;
```

```
                     001A   124          .SBTTL  OUTPUT BLANK CHARACTER
                     001A   125 ;+
                     001A   126 ; EXE$OUTBLANK - OUTPUT BLANK CHARACTER
                     001A   127 ;
                     001A   128 ; THIS ROUTINE IS CALLED VIA A JSB TO OUTPUT A BLANK CHARACTER TO A
                     001A   129 ; SPECIFIED DEVICE.
                     001A   130 ;
                     001A   131 ; INPUTS:
                     001A   132 ;
                     001A   133 ;       R11 = OUTPUT DEVICE CSR ADDRESS.
                     001A   134 ;
                     001A   135 ; OUTPUTS:
                     001A   136 ;
                     001A   137 ;       BLANK CHARACTER IS OUTPUT TO SPECIFIED DEVICE.
                     001A   138 ;-
                     001A   139
                     001A   140 EXE$OUTBLANK::                          ;OUTPUT BLANK CHARACTER
 50   20   9A        001A   141          MOVZBL  #BLANK,R0               ;SET BLANK CHARACTER
      00   11        001D   142          BRB     EXE$OUTCHAR             ;
```

```
                     001F   144          .SBTTL  OUTPUT CHARACTER
                     001F   145 ;+
                     001F   146 ; EXE$OUTCHAR - OUTPUT CHARACTER
                     001F   147 ;
                     001F   148 ; THIS ROUTINE IS CALLED VIA A JSB TO OUTPUT A CHARACTER TO A SPECIFIED
                     001F   149 ; DEVICE.
                     001F   150 ;
                     001F   151 ; INPUTS:
                     001F   152 ;
                     001F   153 ;       R0 = CHARACTER TO OUTPUT.
                     001F   154 ;       R11 = OUTPUT DEVICE CSR ADDRESS (ZERO IMPLIES CONSOLE TERMINAL).
                     001F   155 ;
                     001F   156 ; OUTPUTS:
                     001F   157 ;
                     001F   158 ;       CHARACTER IS OUTPUT TO THE SPECIFIED DEVICE. IF THE CHARACTER
                     001F   159 ;       IS A CARRIAGE RETURN AND THE OUTPUT DEVICE IS THE CONSOLE TERMINAL,
                     001F   160 ;       THEN A SUFFICIENT NUMBER OF FILL CHARACTERS ARE ALSO OUTPUT.
                     001F   161 ;-
                     001F   162
                     001F   163 EXE$OUTCHAR::                           ;OUTPUT CHARACTER
          51    DD   001F   164          PUSHL   R1                     ; GET A WORKING REGISTER
          5B    D5   0021   165          TSTL    R11                    ;CONSOLE TERMINAL?
          08    12   0023   166          BNEQ    20$                    ;IF NEQ NO
00000000'9F    16   0025   167          JSB     @#CON$PUTCHAR          ;CALL CONSOLE TERMINAL SPECIFIC ROUTINE
          0B    11   002B   168          BRB     30$                    ;
6B  0080  8F    B3   002D   169 20$:     BITW    #^X080,TCSR(R11)       ;DEVICE READY?
          F9    13   0032   170          BEQL    20$                    ;IF EQL NO
  02 AB   50    90   0034   171          MOVB    R0,TDBR(R11)           ;OUTPUT CHARACTER
     50   0D    91   0038   172 30$:     CMPB    #CR,R0                 ;CARRIAGE RETURN?
          0F    12   003B   173          BNEQ    60$                    ;IF NEQ NO
          5B    D5   003D   174          TSTL    R11                    ;CONSOLE TERMINAL?
          0B    12   003F   175          BNEQ    60$                    ;IF NEQ NO
          50    D4   0041   176 40$:     CLRL    R0                     ;SET FILL CHARACTER
          02    DD   0043   177          PUSHL   #2                     ;SET FILL COUNT
          D8    10   0045   178 50$:     BSBB    EXE$OUTCHAR            ;OUTPUT A FILL CHARACTER
FB        6E    F5   0047   179          SOBGTR  (SP),50$               ;ANY MORE FILLS TO OUTPUT?
          8E    D5   004A   180          TSTL    (SP)+                  ;CLEAN STACK
          51 8ED0    004C   181 60$:     POPL    R1                     ; RESTORE WORKING REGISTER
             05    004F   182          RSB
                   0050   183
```

```
                     0050   185          .SBTTL  OUTPUT CARRIAGE RETURN/LINE FEED PAIR
                     0050   186 ;+
                     0050   187 ; EXE$OUTCRLF - OUTPUT CARRIAGE RETURN/LINE FEED PAIR
                     0050   188 ;
                     0050   189 ; THIS ROUTINE IS CALLED TO OUTPUT A CARRIAGE RETURN/LINE FEED PAIR TO A
                     0050   190 ; SPECIFIED DEVICE.
                     0050   191 ;
                     0050   192 ; INPUTS:
                     0050   193 ;
                     0050   194 ;       R11 = OUTPUT DEVICE CSR ADDRESS.
                     0050   195 ;
                     0050   196 ; OUTPUTS:
                     0050   197 ;
                     0050   198 ;       A CARRIAGE RETURN FOLLOWED BY A LINE FEED IS OUTPUT TO THE SPECIFIED
                     0050   199 ;       DEVICE.
                     0050   200 ;-
                     0050   201
                     0050   202 EXE$OUTCRLF::                            ;OUTPUT CARRIAGE RETURN/LINE FEED PAIR
 50   0D   9A   0050   203          MOVZBL  #CR,R0                      ;SET CARRIAGE RETURN CHARACTER
      CA   10   0053   204          BSBB    EXE$OUTCHAR                 ;OUTPUT CARRIAGE RETURN
 50   0A   9A   0055   205          MOVZBL  #LF,R0                      ;SET LINE FEED CHARACTER
      C5   11   0058   206          BRB     EXE$OUTCHAR                 ;OUTPUT LINE FEED CHARACTER
```

```
                     005A    208          .SBTTL  OUTPUT COUNTED AND ZERO TERMINATED STRINGS
                     005A    209 ;+
                     005A    210 ; EXE$OUTCSTRING - OUTPUT COUNTED STRING
                     005A    211 ;
                     005A    212 ; THIS ROUTINE IS CALLED VIA A JSB TO OUTPUT A STRING WHOSE FIRST CHARACTER
                     005A    213 ; IS THE NUMBER OF CHARACTERS TO BE OUTPUT.
                     005A    214 ;
                     005A    215 ; INPUTS:
                     005A    216 ;
                     005A    217 ;       R1 = ADDRESS OF COUNTED STRING.
                     005A    218 ;       R11 = OUTPUT DEVICE CSR ADDRESS.
                     005A    219 ;
                     005A    220 ; OUTPUTS:
                     005A    221 ;
                     005A    222 ;       THE NUMBER OF CHARACTERS SPECIFIED BY THE FIRST BYTE OF THE STRING
                     005A    223 ;       ARE OUTPUT TO THE SPECIFIED DEVICE. OUTPUT IS ALSO TERMINATED IF
                     005A    224 ;       A ZERO BYTE IS ENCOUNTERED BEFORE THE COUNT IS EXHAUSTED.
                     005A    225 ;-
                     005A    226
                     005A    227          .ENABL  LSB
                     005A    228 EXE$OUTCSTRING::                        ;OUTPUT COUNTED STRING
 52    81    9A      005A    229          MOVZBL  (R1)+,R2               ;SET COUNT OF CHARACTERS TO OUTPUT
       04    11      005D    230          BRB     10$                    ;
                     005F    231
                     005F    232 ;+
                     005F    233 ; EZE$OUTZSTRING - OUTPUT ZERO TERMINATED STRING
                     005F    234 ;
                     005F    235 ; THIS ROUTINE IS CALLED VIA A JSB TO OUTPUT A STRING THAT IS TERMINATED BY
                     005F    236 ; A ZERO BYTE.
                     005F    237 ;
                     005F    238 ; INPUTS:
                     005F    239 ;
                     005F    240 ;       R1 = ADDRESS OF ZERO TERMINATED STRING.
                     005F    241 ;       R11 = OUTPUT DEVICE CSR ADDRESS.
                     005F    242 ;
                     005F    243 ; OUTPUTS:
                     005F    244 ;
                     005F    245 ;       CHARACTERS FROM THE SPECIFIED STRING ARE OUTPUT UNTIL A ZERO BYTE
                     005F    246 ;       IS ENCOUNTERED.
                     005F    247 ;-
                     005F    248
                     005F    249 EXE$OUTZSTRING::                        ;OUTPUT ZERO TERMINATED STRING
 52    FF  8F    9A      005F    250          MOVZBL  #255,R2                ;SET MAXIMUM ALLOWABLE STRING LENGTH
    50     81    9A      0063    251 10$:     MOVZBL  (R1)+,R0               ;GET NEXT CHARACTER FROM INPUT STRING
           05    13      0066    252          BEQL    20$                    ;IF EQL END OF STRING
           B5    10      0068    253          BSBB    EXE$OUTCHAR            ;OUTPUT CHARACTER
        F6 52    F5      006A    254          SOBGTR  R2,10$                 ;ANY MORE CHARACTERS TO OUTPUT?
                 05      006D    255 20$:     RSB                            ;
                         006E    256          .DSABL  LSB
                         006E    257
                         006E    258          .END
```

```
BLANK           = 00000020
CON$PUTCHAR       ********  X   03
CONTROL_Q       = 00000011
CONTROL_S       = 00000013
CR              = 0000000D
EXE$AB_HEXTAB     00000000 RG    02
EXE$OUTBLANK      0000001A RG    03
EXE$OUTCHAR       0000001F RG    03
EXE$OUTCRLF       00000050 RG    03
EXE$OUTCSTRING    0000005A RG    03
EXE$OUTHEX        00000000 RG    03
EXE$OUTZSTRING    0000005F RG    03
LF              = 0000000A
RCSR            = FFFFFFFC
RDBR            = FFFFFFFE
SEVEN           = 00000037
TCSR            = 00000000
TDBR            = 00000002
ZERO            = 00000030
```

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	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$AEXENONPAGED	00000010 (16.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
Z$INIT__BUGC	0000006E (110.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.08	00:00:01.26
Command processing	128	00:00:00.52	00:00:03.05
Pass 1	138	00:00:01.53	00:00:06.43
Symbol table sort	0	00:00:00.09	00:00:00.23
Pass 2	59	00:00:00.60	00:00:01.67
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	1	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	368	00:00:02.87	00:00:12.70

```
The working set limit was 1050 pages.
7334 bytes (15 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 99 non-local and 8 local symbols.
258 source lines were read in Pass 1, producing 15 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	4

139 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:CONSOLIO/OBJ=OBJ$:CONSOLIO MSRC$:CONSOLIO/UPDATE=(ENH$:CONSOLIO)+EXECML$/LIB
```

0373 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CMODSSDSP
LIS

BUGCHECK
LIS

COMDRVSUB
LIS

CLUSTRVEC
LIS

BUFFERCTL
LIS

DEADLOCK
LIS

CVTFILNAM
LIS

BOOPARAM
LIS

CJFSYSVEC
LIS

CVTATB
LIS

BUGCHKMSG
LIS

CONSOLIO
LIS