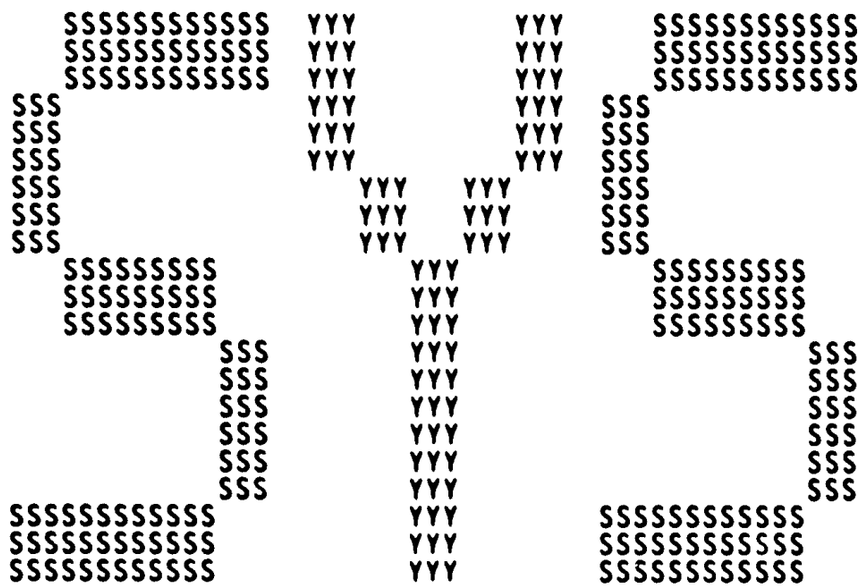

FILEID**RMSRESET

SY
VO

```
RRRRRRRR    MM      MM    SSSSSSSS  RRRRRRRR    EEEEEEEEEE    SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT
RRRRRRRR    MM      MM    SSSSSSSS  RRRRRRRR    EEEEEEEEEE    SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT
RR      RR  MMMM  MMMM  SS          RR      RR  EE          SS          EE              TT
RR      RR  MMMM  MMMM  SS          RR      RR  EE          SS          EE              TT
RR      RR  MM  MM  MM  SS          RR      RR  EE          SS          EE              TT
RR      RR  MM  MM  MM  SS          RR      RR  EE          SS          EE              TT
RRRRRRRR    MM      MM    SSSSSS    RRRRRRRR    EEEEEEEE      SSSSSS    EEEEEEEE        TT
RRRRRRRR    MM      MM    SSSSSS    RRRRRRRR    EEEEEEEE      SSSSSS    EEEEEEEE        TT
RR  RR      MM      MM          SS  RR  RR      EE                  SS  EE              TT
RR  RR      MM      MM          SS  RR  RR      EE                  SS  EE              TT
RR    RR    MM      MM          SS  RR    RR    EE                  SS  EE              TT          ....
RR    RR    MM      MM          SS  RR    RR    EE                  SS  EE              TT          ....
RR      RR  MM      MM  SSSSSSSS    RR      RR  EEEEEEEEEE  SSSSSSSS    EEEEEEEEEE      TT          ....
RR      RR  MM      MM  SSSSSSSS    RR      RR  EEEEEEEEEE  SSSSSSSS    EEEEEEEEEE      TT          ....

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              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  RMSRESET - RMS IMAGE I/O SEGMENT REINITIALIZE ROUTINE
0000     2          .IDENT  'V04-000'
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26
0000    27 ;++
0000    28 ; FACILITY:     EXECUTIVE, IMAGE ACTIVE AND RESET ROUTINES
0000    29 ;
0000    30 ; ABSTRACT:     REINITIALIZES RMS IMAGE IO SEGMENT LIST HEADERS.
0000    31 ;
0000    32 ; ENVIRONMENT:  KERNEL MODE.
0000    33 ;
0000    34 ;--
0000    35 ;
0000    36          .SBTTL  HISTORY                     ; DETAILED
0000    37 ;
0000    38 ; AUTHOR: LEO LAVERDURE , CREATION DATE: 21-FEB-78
0000    39 ;
0000    40 ; MODIFIED BY:
0000    41 ;
0000    42 ;       V03-002 WMC0001         Wayne Cardoza           19-Mar-1984
0000    43 ;               RMSSET routine to initialize the image I/O area.
0000    44 ;
0000    45 ;       V03-001 SRB0111         Steve Beckhardt         10-Feb-1984
0000    46 ;               Added blocking AST routine for RMS directory cache
0000    47 ;               invalidate.
0000    48 ;
0000    49 ;--
0000    50          .SBTTL  DECLARATIONS
0000    51
0000    52 ;
0000    53 ; INCLUDE FILES:
0000    54 ;
0000    55          $DYNDEF                         ;DATA STRUCTURE TYPE CODES
0000    56          $IHDDEF                         ;IMAGE HEADER DEFINITIONS
0000    57          $IMPDEF                         ;RMS IMPURE AREA DEFINITIONS
```

```
0000     58          $SSDEF                          ;SYSTEM STATUS DEFINITIONS
0000     59          $UCBDEF                         ;UCB DEFINITIONS
0000     60
0000     61 ;
0000     62 ; EXTERNAL SYMBOLS:
0000     63 ;
0000     64
0000     65 ;
0000     66 ; MACROS:
0000     67 ;
0000     68 ;
0000     69 ; EQUATED SYMBOLS:
0000     70 ;
0000     71 ;
0000     72 ; OWN STORAGE:
0000     73 ;
```

```
                        .SBTTL  RMSRESET - IMAGE I/O SEGMENT RESET

                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ; CALLING SEQUENCE:
                ;
                ;       BSBW    RMS$RESET
                ;               MAY BE CALLED FROM EITHER EXEC OR KERNEL MODE
                ;
                ; INPUT PARAMETERS:
                ;
                ; IMPLICIT INPUTS:
                ;
                ;       NONE
                ;
                ; OUTPUT PARAMETERS:
                ;
                ;       RESETS THE IMAGE I/O FREE PAGE LIST TO EMPTY
                ;       RESETS THE IMAGE I/O FREE SPACE LIST TO EMPTY
                ;       ZEROES THE IMAGE I/O SEGEMENT BASE ADDRESS AND LENGTH
                ;       ZEROES THE IMAGE I/O IFAB AND IRAB TABLE LINK ADDRESSES
                ;       IF NO ACTIVE IMAGE I/O IFABS, EXITS WITH SUCCESS, ELSE
                ;               ZEROES THE # OF ACTIVE IFABS
                ;               ZEROES THE IMAGE I/O IFAB AND IRAB TABLES
                ;               EXIT WITH STATUS SS$_MUSTCLOSEFL
                ;
                ;       R0 = ERROR STATUS CODE
                ;       R1,R2 ARE DESTROYED
                ;
                ; IMPLICIT OUTPUTS:
                ;       NONE
                ;
                ; COMPLETION CODES:
                ;       SEE ABOVE
                ;
                ; SIDE EFFECTS:
                ;
                ;       IF AN ERROR IS RETURNED, THE IMAGE I/O FILES CANNOT BE RUN DOWN.
                ;       THIS IS OF LITTLE CONSEQUENCE AS THIS IS AN ERROR CONDITION
                ;       ARISING FROM THE CLI'S FAILURE TO DO THE RUNDOWN BEFORE STARTING
                ;       ANOTHER IMAGE.  THE IMAGE MAY BE STARTED.
                ;
                ;--

                ;*****************************************************************************

                ;*************** THE FOLLOWING CODE MAY BE PAGED *****************

00000000                .PSECT  Y$EXEPAGED

                ;*****************************************************************************

                RMS$RESET::
```

```
                00000000'9F   DE  0000   132          MOVAL   @#PIO$GL_IIOFSPLH,R2     ;RMS FREE PAGE LIST HEAD
                      62   52   D0  0007   133          MOVL    R2,(R2)                ;MAKE THIS LIST EMPTY
                   04 A2   52   D0  000A   134          MOVL    R2,4(R2)               ;TAIL POINTS AT HEAD
                   000C'C2      DE  000E   135          MOVAL   W^<PIO$GW_IIOIMPA -    ;FORM ADDRESS OF IMAGE I/O
                           52       0012   136                  +IMP$L_FREEPGLH-PIO$GL_IIOFSPLH>(R2),R2 ;FREE PAGE LIST
                      62   52   D0  0013   137          MOVL    R2,(R2)                ;MAKE THE LIST EMPTY
                   04 A2   52   D0  0016   138          MOVL    R2,4(R2)               ;TAIL POINTS AT HEAD
                                    001A   139
                                    001A   140          ASSUME  IMP$L_IOSEGLEN EQ IMP$L_IOSEGADDR+4
                      F8 A2   7C   001A   141          CLRQ    <IMP$L_IOSEGADDR-IMP$L_FREEPGLH>(R2)    ;NO IMAGE I/O SEGMENT
                      0C B2   D4   001D   142          CLRL    @IMP$L_IFABTBL-IMP$L_FREEPGLH(R2)       ;ZERO IFAB TABLE LINK
                      10 B2   D4   0020   143          CLRL    @IMP$L_IRABTBL-IMP$L_FREEPGLH(R2)       ;ZERO IRAB TABLE LINK
                      16 A2   B5   0023   144          TSTW    IMP$W_NUM_IFABS-IMP$L_FREEPGLH(R2)      ;ANY ACTIVE IFABS?
                         04   12   0026   145          BNEQ    10$                                     ;BRANCH IF NOT
                      50   01   D0   0028   146          MOVL    #1,R0                                   ;SHOW SUCCESS
                                    002B   147          RSB                                             ;BACK TO CALLER
                                    002C   148 ;
                                    002C   149 ;  I/O RUNDOWN NOT DONE!!
                                    002C   150 ;
                                    002C   151 ;  JUST CLEAR THE IFAB AND IRAB TABLE POINTERS SO THAT RMS CAN CONTINUE.
                                    002C   152 ;
                                    002C   153 ;  (ASSUMES THAT THE IFAB TABLE IS IMMEDIATELY FOLLOWED BY THE IRAB TABLE.)
                                    002C   154 ;
                      16 A2   B4   002C   155 10$:     CLRW    IMP$W_NUM_IFABS-IMP$L_FREEPGLH(R2)      ;SAY NO ACTIVE IFABS
                   52 0C A2   D0   002F   156          MOVL    IMP$L_IFABTBL-IMP$L_FREEPGLH(R2),R2     ;GET ADDR OF IFAB TABLE
                      51   10   D0   0033   157          MOVL    #IMP$C_ENTPERSEG+1,R1                   ; # OF QUADWORDS TO ZERO
                           82   7C   0036   158 20$:     CLRQ    (R2)+                                   ;ZERO IFAB AND IRAB ADDRS
                      FB   51   F5   0038   159          SOBGTR  R1,20$                                  ;LOOP
                50   0948   8F   3C   003B   160          MOVZWL  #SS$_MUSTCLOSEFL,R0                     ;FAILURE STATUS
                                    05   0040   161          RSB
```

```
                     0041   163          .SBTTL  RM$SET - INITIALIZE IMAGE I/O SEGMENT
                     0041   164
                     0041   165 ;++
                     0041   166 ; FUNCTIONAL DESCRIPTION:
                     0041   167 ;
                     0041   168 ;
                     0041   169 ; CALLING SEQUENCE:
                     0041   170 ;
                     0041   171 ;       BSBW    RM$SET
                     0041   172 ;               MAY BE CALLED FROM EITHER EXEC OR KERNEL MODE
                     0041   173 ;
                     0041   174 ;
                     0041   175 ; INPUT PARAMETERS:
                     0041   176 ;
                     0041   177 ;       R0 =  IMAGE HEADER LINK FLAGS
                     0041   178 ;       R1 -> DESCRIPTOR OF EXISTING IMAGE I/O SEGMENT SPACE
                     0041   179 ;
                     0041   180 ; IMPLICIT INPUTS:
                     0041   181 ;
                     0041   182 ;       NONE
                     0041   183 ;
                     0041   184 ; OUTPUT PARAMETERS:
                     0041   185 ;
                     0041   186 ;       R0,R1,R2 ARE DESTROYED
                     0041   187 ;
                     0041   188 ; IMPLICIT OUTPUTS:
                     0041   189 ;       NONE
                     0041   190 ;
                     0041   191 ; COMPLETION CODES:
                     0041   192 ;       NONE
                     0041   193 ;
                     0041   194 ; SIDE EFFECTS:
                     0041   195 ;
                     0041   196 ;       NONE
                     0041   197 ;--
                     0041   198
                     0041   199 ;
                     0041   200 ; ******************************************************************************
                     0041   201 ;
                     0041   202 ; ***************** THE FOLLOWING CODE MAY BE PAGED *****************
                     0041   203 ;
                     00000041   204          .PSECT  Y$EXEPAGED
                     0041   205 ;
                     0041   206 ; ******************************************************************************
                     0041   207 ;
                     0041   208 RM$SET::
    52   00000000'EF   DE   0041   209          MOVAL   PIO$GW_IIOIMPA,R2               ;IMAGE I/O IMPURE AREA
          00 62    05   E5   0048   210          BBCC    #IMP$V_NOPOBUFS,(R2),5$         ;ASSUME NO PO BUFFERS ALLOWED
          04 50    02   E1   004C   211 5$:      BBC     #IHD$V_NOPOBUFS,R0,10$          ;CHECK LINK FLAGS
          00 62    05   E2   0050   212          BBSS    #IMP$V_NOPOBUFS,(R2),10$        ;NONE ALLOWED
08 A2   00000000'EF   D0   0054   213 10$:     MOVL    PIO$GQ_IIODEFAULT,IMP$L_IOSEGLEN(R2) ;DEFAULT I/O SEGMENT LENGTH
04 A2   00000004'EF   D0   005C   214          MOVL    PIO$GQ_IIODEFAULT+4,IMP$L_IOSEGADDR(R2) ;SEGMENT ADDRESS
                61   D5   0064   215          TSTL    (R1)                            ;IS AN ADDITIONAL SEGMENT SPECIFIED
                0C   13   0066   216          BEQL    20$
          50    04   A1   D0   0068   217          MOVL    4(R1),R0                        ;ADDITIONAL AREA
          08 A0    61   D0   006C   218          MOVL    (R1),8(R0)                      ;LENGTH OF SEGMENT
          0C A2    60   0E   0070   219          INSQUE  (R0),IMP$L_FREEPGLH(R2)         ;PUT IT ON FREE LIST
```

```
 05  0074    220 20$:    RSB
```

```
                     0075   222          .SBTTL  RM$DIRCACHE_BLKAST - DIRECTORY CACHE BLOCKING AST
                     0075   223 ;++
                     0075   224 ; FUNCTIONAL DESCRIPTION:
                     0075   225 ;
                     0075   226 ;       THIS ROUTINE IS THE SYSTEM BLOCKING AST ROUTINE FOR THE RMS DIRECTORY
                     0075   227 ;       CACHE. IT IS LOCATED HERE BECAUSE IT IS CALLED AT IPL$_SYNCH.
                     0075   228 ;       THIS ROUTINE BUMPS THE DIRECTORY SEQUENCE COUNTER IN THE UCB
                     0075   229 ;       AND CLEARS THE CACHE ARMED BIT.  THEN THE FIRST RMS TO NOTICE
                     0075   230 ;       THE CACHE ISN'T VALID WILL DO A LOCK CONVERSION TO REARM THE LOCK.
                     0075   231 ;
                     0075   232 ; CALLING SEQUENCE:
                     0075   233 ;
                     0075   234 ;       JSB     RM$DIRCACHE_BLKAST AT IPL$_SYNCH
                     0075   235 ;
                     0075   236 ; INPUT PARAMETERS:
                     0075   237 ;
                     0075   238 ;       R1      ADDRESS OF UCB
                     0075   239 ;
                     0075   240 ; OUTPUT PARAMETERS:
                     0075   241 ;
                     0075   242 ;       NONE
                     0075   243 ;
                     0075   244 ; SIDE EFFECTS:
                     0075   245 ;
                     0075   246 ;       NONE
                     0075   247 ;--
                     0075   248
                00000000    249          .PSECT  AEXENONPAGED
                     0000   250
                     0000   251 RM$DIRCACHE_BLKAST::
          10    91 0000    252          CMPB    #DYN$C_UCB,-            ; MAKE SURE WE ARE POINTING
        0A  A1       0002   253                  UCB$B_TYPE(R1)          ; AT A UCB
              0C   12 0004    254          BNEQ    10$
        00AC  C1   B6 0006    255          INCW    UCB$W_DIRSEQ(R1)        ; INCREMENT SEQ. #
        8000  8F   AA 000A    256          BICW    #UCB$M_AST_ARMED,-      ; INDICATE AST IS NO LONGER ARMED
        00AC  C1       000E   257                  UCB$W_DIRSEQ(R1)
              05   00 0011    258          RSB
                     0012   259
                     0012   260 10$:     BUG_CHECK       NOTUCBUCB,FATAL
                     0016   261
                     0016   262
                     0016   263
                     0016   264
                     0016   265          .END
```

```
BUG$_NOTUCBUCB                  ********    X   03
DYN$C_UCB                     = 00000010
IHD$V_NOPOBUFS                = 00000002
IMP$C_ENTPERSEG               = 0000000F
IMP$L_FREEPGLH                = 0000000C
IMP$L_IFABTBL                 = 00000018
IMP$L_IOSEGADDR               = 00000004
IMP$L_IOSEGLEN                = 00000008
IMP$L_IRABTBL                 = 0000001C
IMP$V_NOPOBUFS                = 00000005
IMP$W_NUM_IFABS               = 00000022
PIO$GL_IIOFSPLH                 ********    X   02
PIO$GQ_IIODEFAULT               ********    X   02
PIO$GW_IIOIMPA                  ********    X   02
RM$DIRCACHE_BLKAST              00000000 RG     03
RM$RESET                        00000000 RG     02
RM$SET                          00000041 RG     02
SS$_MUSTCLOSEFL               = 00000948
UCB$B_TYPE                    = 0000000A
UCB$M_AST_ARMED               = 00008000
UCB$W_DIRSEQ                  = 000000AC
```

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
Y$EXEPAGED	00000075 (117.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
AEXENONPAGED	00000016 (22.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.03	00:00:01.40
Command processing	106	00:00:00.50	00:00:03.19
Pass 1	291	00:00:08.54	00:00:28.79
Symbol table sort	0	00:00:01.44	00:00:03.51
Pass 2	59	00:00:01.56	00:00:07.77
Symbol table output	4	00:00:00.04	00:00:00.55
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	494	00:00:12.13	00:00:45.24

```
The working set limit was 1350 pages.
48710 bytes (96 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 993 non-local and 6 local symbols.
265 source lines were read in Pass 1, producing 15 object records in Pass 2.
14 pages of virtual memory were used to define 13 macros.
```

```
                                  +---------------------------+
                                  ! Macro library statistics !
                                  +---------------------------+

Macro library name                 Macros defined
------------------                 --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                5
_$255$DUA28:[SYSLIB]STARLET.MLB;2             5
TOTALS (all libraries)                       10

1075 GETS were required to define 10 macros.

There were no errors, warnings or information messages.

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

0379 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RELOCDRV
LIS

RMSRESET
LIS

PROCSTRT
LIS

PHDUTL
LIS

PMSDAT
LIS

POSTEF
LIS

POWERFAIL
LIS

PRDEF
LIS

PTEDUMP
LIS

PDAT
LIS

RMSVECTOR
LIS