```
HHH         HHH  LLL              DDDDDDDDDDDD
HHH         HHH  LLL              DDDDDDDDDDDDD
HHH         HHH  LLL              DDDDDDDDDDDDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHHHHHHHHHHHHHHH  LLL              DDD         DDD
HHHHHHHHHHHHHHHH  LLL              DDD         DDD
HHHHHHHHHHHHHHHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLLLLLLLLLLLLLL  DDDDDDDDDDDD
HHH         HHH  LLLLLLLLLLLLLLL  DDDDDDDDDDDD
HHH         HHH  LLLLLLLLLLLLLLL  DDDDDDDDDDDD
HHH         HHH  LLLLLLLLLLLLLLL  DDDDDDDDDDDD
```

```
HH      HH  LL          DDDDDDD    MM      MM    AAAAAA     IIIIII    NN      NN
HH      HH  LL          DDDDDDDD   MM      MM    AAAAAA     IIIIII    NN      NN
HH      HH  LL          DD      DD MMMM  MMMM  AA      AA      II      NN      NN
HH      HH  LL          DD      DD MMMM  MMMM  AA      AA      II      NN      NN
HH      HH  LL          DD      DD MM  MM  MM  AA      AA      II      NNNN    NN
HH      HH  LL          DD      DD MM  MM  MM  AA      AA      II      NNNN    NN
HHHHHHHHHH  LL          DD      DD MM      MM  AA      AA      II      NN  NN  NN
HHHHHHHHHH  LL          DD      DD MM      MM  AA      AA      II      NN  NN  NN
HH      HH  LL          DD      DD MM      MM  AAAAAAAAAA      II      NN    NNNN
HH      HH  LL          DD      DD MM      MM  AAAAAAAAAA      II      NN    NNNN
HH      HH  LL          DD      DD MM      MM  AA      AA      II      NN      NN    ....
HH      HH  LL          DD      DD MM      MM  AA      AA      II      NN      NN    ....
HH      HH  LLLLLLLLLL  DDDDDDDD   MM      MM  AA      AA    IIIIII    NN      NN    ....
HH      HH  LLLLLLLLLL  DDDDDDD    MM      MM  AA      AA    IIIIII    NN      NN    ....

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  HLDMAIN - HLD MAINLINE
0000     2          .IDENT  'V04-000'
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: DECNET HOST LOADER (HLD)
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;
0000    34  ;       HLD IS A COMPONENT OF DECNET/VAX-11. IT PROVIDES ACCESS TO
0000    35  ;       RSX11S TASK IMAGES STORED ON A VAX SYSTEM.
0000    36  ;
0000    37  ; ENVIRONMENT:
0000    38  ;
0000    39  ;       THE HLD IMAGE EXECUTES IN THE CONTEXT OF A PROCESS CREATED BY
0000    40  ;       NETACP. IT RUNS IN USER MODE AND REQUIRES NETWORK PRIVILEGE.
0000    41  ;
0000    42  ; AUTHOR: SCOTT G. DAVIS,       CREATION DATE: 10-MAY-79
0000    43  ;
0000    44  ; MODIFICATIONS:
0000    45  ;
0000    46  ;       V001    SGD0003         19-Mar-1984
0000    47  ;               Cleanup log on normal termination
0000    48  ;--
0000    49  ;
0000    50  ; INCLUDE FILES
0000    51  ;
0000    52  ;
0000    53  ; MACROS:
0000    54  ;
0000    55  ;       NONE
```

```
                     0000     57          .SBTTL  HLD$START - MAINLINE
                 00000000     58          .PSECT  HLD$CODE        NOSHR,EXE,RD,NOWRT
                     0000     59
                     0000     60 ;++
                     0000     61 ; FUNCTIONAL DESCRIPTION:
                     0000     62 ;
                     0000     63 ;       HLD$START IS RESPONSIBLE FOR HLD INITIALIZATION AND TERMINATION.
                     0000     64 ;
                     0000     65 ; CALLING SEQUENCE:
                     0000     66 ;
                     0000     67 ;       HLD IS INITIATED BY NETACP.
                     0000     68 ;
                     0000     69 ; INPUT PARAMETERS:
                     0000     70 ;
                     0000     71 ;       NONE
                     0000     72 ;
                     0000     73 ; IMPLICIT INPUTS:
                     0000     74 ;
                     0000     75 ;       LOGICAL NAME 'SYS$NET'
                     0000     76 ;
                     0000     77 ;--
                     0000     78
                0000 0000     79          .ENTRY  HLD$START,^M<>          ; ENTRY POINT FROM EXECUTIVE
                     0002     80
                     0002     81 ;
                     0002     82 ; OPEN THE PRINT FILE
                     0002     83 ;
                     0002     84
          50 0000'CF   9E 0002     85          MOVAB   W^HLD$PRTFAB,R0         ; SET TO OPEN FILE
             FFF6'     30 0007     86          BSBW    HLD$RMS_OPEN            ; DO IT
          50 0000'CF   9E 000A     87          MOVAB   W^HLD$PRTRAB,R0         ; SET TO CONNECT REACORD STREAM
             FFEE'     30 000F     88          BSBW    HLD$RMS_CONNECT         ; DO IT
                     0012     89 ;
                     0012     90 ; CREATE A CONTROL/INFORMATION PATH TO NETACP IN PREPARATION FOR
                     0012     91 ; NON-TRANSPARENT NETWORK I/O. ALSO ASSOCIATE A MAILBOX WITH THE
                     0012     92 ; CHANNEL.
                     0012     93 ;
                     0012     94
                     0012     95          $ASSIGN_S DEVNAM=W^HLD$GQ_LNKNAM- ; "NET0:" REFERS TO NETACP
                     0012     96                    CHAN=W^HLD$GW_LNKCHN    ; STORE CHANNEL #
                     0023     97          CHECK_SS                        ; CHECK STATUS CODE
                     0026     98
                     0026     99 ;
                     0026    100 ; TRANSLATE THE LOGICAL NAME 'SYS$NET'. ITS EQUIVALENCE STRING IS
                     0026    101 ; SET-UP BY NETACP TO BE THE ENTIRE NETWORK CONNECT BLOCK (NCB) WHICH
                     0026    102 ; IS REQUIRED TO GAIN NON-TRANSPARENT ACCESS TO THE NETWORK.
                     0026    103 ;
                     0026    104
          57 0000'CF   7E 0026    105          MOVAQ   W^HLD$GQ_NCBDESC,R7     ; GET ADDRESS OF NCB DESCRIPTOR
             67   40 8F   9A 002B    106          MOVZBL  #64,(R7)                ; STORE BUFFER SIZE
       04 A7 0000'CF   9E 002F    107          MOVAB   W^HLD$GT_NCBBUF,4(R7)   ; STORE BUFFER ADDRESS
                     0035    108          $TRNLOG_S LOGNAM=W^HLD$GQ_SYSNAM- ; ADDRESS OF LOGICAL NAME DESCRIPTOR
                     0035    109                    RSLLEN=(R7)-             ; UPDATE SIZE IN NCB DESCRIPTOR
                     0035    110                    RSLBUF=(R7)              ; PUT EQUIVALENCE STRING IN NCB
                     004A    111          CHECK_SS                        ; CHECK STATUS CODE
                     004D    112
                     004D    113 ;++
```

```
                        004D   114 ; THE GENERAL FORMAT OF THE EQUIVALENCE STRING DERIVED FROM SYS$NET IS:
                        004D   115 ;
                        004D   116 ;       nodespec::"objecttype=taskid/netacp_string"
                        004D   117 ;       userdata (before close ") consists of:
                        004D   118 ;               4 bytes of rad50 task name
                        004D   119 ;               1 word of partition address (0=>mapped)
                        004D   120 ;               1 word of partition size (in bytes or 32.-word blocks)
                        004D   121 ;               1 byte of lun (general purpose) flag - 0=no,1=>lun fixing
                        004D   122 ;               1 byte of function code (-1=overlay,0=read, 1=chkread, 2=chkwrt)
                        004D   123 ;               2 words of VBN if overlay request
                        004D   124 ;
                        004D   125 ; On return, the optional data consists of:
                        004D   126 ;               1 word of transfer size
                        004D   127 ;               1 byte of no. of luns to fix
                        004D   128 ;
                        004D   129 ; THE FOLLOWING WILL BE OUTPUT TO THE PRINT FILE:
                        004D   130 ;
                        004D   131 ;       nodespec::"objecttype=taskid"
                        004D   132 ;
                        004D   133 ; THE FOLLOWING WILL BE PUT INTO THE NCB FOR A CONNECT ACCEPT
                        004D   134 ;
                        004D   135 ;       nodespec::"objecttype=taskid/next_two_bytes"
                        004D   136 ; userdata consists of actual transfer size in place of partition size
                        004D   137 ;--
                        004D   138
      04 B7   67   3A   3A  004D   139         LOCC    #^A\:\,(R7),@4(R7)      ; FIND NODE DELIMITER
                     11   13  0052   140         BEQL    20$                     ; IF EQL NOT FOUND
0000'CF   67   50   A3   0054   141         SUBW3   R0,(R7),W^HLD$GQ_NODEDESC ; STORE NAME LENGTH
    0000'CF   57   D0   005A   142         MOVL    R7,W^HLD$GL_IOPARAM2    ; ADDRESS OF NCB DESCRIPTOR
    61   50   2F   3A   005F   143         LOCC    #^A\/\,R0,(R1)          ; FIND TASKID DELIMITER
                     07   12  0063   144         BNEQ    30$                     ; BRANCH IF SLASH FOUND
                          0065   145 20$:
    50   0000'8F   3C   0065   146         MOVZWL  #SS$_IVDEVNAM,R0        ; MAKE THIS THE EXIT ERROR STATUS
                     64   11  006A   147         BRB     HLD$EXIT_TO_VMS         ; FATAL ERROR!!
                          006C   148 ;
                          006C   149 ; RESPOND TO THE CONNECT INITIATE WITH A CONNECT CONFIRM (WITHOUT USERDATA)
                          006C   150 ; TO COMPLETE THE LOGICAL LINK.
                          006C   151 ;
                          006C   152
                          006C   153 30$:
    51   03   C0   006C   154         ADDL    #3,R1                   ; SKIP "/" AND 2 RESERVED BYTES
                          006F   155 ;
                          006F   156 ; NOW VERIFY THE CONNECT AND CHECK THE DATA FILE
                          006F   157 ;
       FF8E'   30   006F   158         BSBW    HLD$GET_FILE            ; SCAN HLD.DAT
       FF8B'   30   0072   159         BSBW    HLD$OPEN_TSKFIL         ; OPEN AND PROCESS THE TASK FILE
                          0075   160 ;
                          0075   161 ;       CONFIRM THE LOGICAL LINK
                          0075   162 ;
       FF88'   30   0075   163         BSBW    HLD$NET_IO              ; CONFIRM THE LINK
                          0078   164 ;
                          0078   165 ; Now fix it so that HLD waits for disconnect if transfer is to host.
                          0078   166 ;
                          0078   167         $SETEF_S        EFN=#2          ; Set event flag as default
00'    0000'CF   B1   0081   168         CMPW    W^HLD$GW_SAVEFUNC,S^#IO$_READVBLK ; Am I reading?
                     1E   12  0086   169         BNEQ    40$                     ; If NEQ no
                          0088   170 ;
```

```
                     0088   171 ; Issue a phony QIO which will terminate when the link is broken
                     0088   172 ;
                     0088   173         $QIO_S  -                               ; ISSUE THE REQUEST
                     0088   174                 CHAN=   W^HLD$GW_LNKCHN- ;
                     0088   175                 FUNC=   S^#IO$_WRITEVBLK -; Do a write
                     0088   176                 EFN=    #2-                     ;
                     0088   177                 P1=     .-
                     0088   178                 P2=     #0
                     00A6   179 40$:
                     00A6   180 ;
                     00A6   181 ; NOW DO THE ACTUAL TASK TRANSFER
                     00A6   182 ;
0000'CF   0000'CF    9E   00A6   183         MOVAB   W^HLD$AB_BUFFER,W^HLD$GL_IOPARAM1 ; SET BUFFER ADDRESS
0000'CF   0000'CF    B0   00AD   184         MOVW    W^HLD$GW_SAVEFUNC,W^HLD$GW_IOFUNC ; SET UP TRANSFER FUNCTION
             28      10   00B4   185         BSBB    DO_IO                   ; DO THE TRANSFER
          58 56      7D   00B6   186         MOVQ    R6,R8                   ; SET UP FOR LUN BLOCK TRANSFER
     0000'CF 02      D0   00B9   187         MOVL    #2,W^HLD$GL_TSKBKT      ; Set up VBN of LUN block(s)
             1E      10   00BE   188         BSBB    DO_IO                   ; SEND THEM, MAYBE
                     00C0   189 ;
                     00C0   190 ; Now wait, if necessary
                     00C0   191 ;
                     00C0   192         $WAITFR_S       EFN=#2
50   00030001 8F      D0   00C9   193         MOVL    #^X30001,R0             ; Force "NORMAL" to the log file
                     00D0   194
                     00D0   195 HLD$EXIT_TO_VMS::                            ;
                50   DD   00D0   196         PUSHL   R0                      ; SAVE EXIT STATUS CODE
              FF2B'  30   00D2   197         BSBW    HLD$PRINT               ; PRINT RESULTS
                     00D5   198         $EXIT_S (SP)+                   ; EXIT TO VMS
```

```
                          00DE   200 ;++
                          00DE   201 ;
                          00DE   202 ; DO_IO - CALL I/O ROUTINES TO DO REAL TRANSFER WORK
                          00DE   203 ;
                          00DE   204 ; INPUTS:
                          00DE   205 ;       R8 - NO. OF INTEGRAL BLOCKS TO TRANSFER
                          00DE   206 ;       R9 - SIZE IN BYTES OF BLOCK AFTER LAST INTEGRAL BLOCK OR 0
                          00DE   207 ;
                          00DE   208 ;--
                          00DE   209
                          00DE   210 DO_IO:
0000'CF   0000'CF   3C    00DE   211          MOVZWL  W^HLD$GW_IOLEN,W^HLD$GL_IOPARAM2 ; LENGTH OF ALL BUT LAST TRANSFER
                    02    11    00E5   212          BRB     50$                      ; CHECK THE "NO FULL BLOCK" CASE
                          00E7   213 40$:
                    0A    10    00E7   214          BSBB    55$                      ; CALL THE I/O ROUTINES
                  FB 58   F4    00E9   215 50$:     SOBGEQ  R8,40$                   ; LOOP
                          00EC   216
                          00EC   217
          0000'CF   59    B0    00EC   218          MOVW    R9,W^HLD$GL_IOPARAM2     ; SIZE OF LAST TRANSFER
                    0C    13    00F1   219          BEQL    60$                      ; IF EQL NOTHING
                          00F3   220 55$:
              0000'DF   16    00F3   221          JSB     @W^HLD$GL_IOROUT_1       ; DO 1ST KIND OF I/O
              0000'DF   16    00F7   222          JSB     @W^HLD$GL_IOROUT_2       ; DO 2ND KIND OF I/O
              0000'CF   D6    00FB   223          INCL    W^HLD$GL_TSKBKT          ; ADVANCE THE BLOCK NO.
                          00FF   224 60$:
                    05    00FF   225          RSB                              ; DONE
                          0100   226
                          0100   227          .END    HLD$START
```

```
$$T1            = 00000001
DO_IO             000000DE R      01
HLD$AB_BUFFER     ********   X    01
HLD$CHECK_SS      ********   X    01
HLD$EXIT_TO_VMS   000000D0 RG     01
HLD$GET_FILE      ********   X    01
HLD$GL_IOPARAM1   ********   X    01
HLD$GL_IOPARAM2   ********   X    01
HLD$GL_IOROUT_1   ********   X    01
HLD$GL_IOROUT_2   ********   X    01
HLD$GL_TSKBKT     ********   X    01
HLD$GQ_LNKNAM     ********   X    01
HLD$GQ_NCBDESC    ********   X    01
HLD$GQ_NODEDESC   ********   X    01
HLD$GQ_SYSNAM     ********   X    01
HLD$GT_NCBBUF     ********   X    01
HLD$GW_IOFUNC     ********   X    01
HLD$GW_IOLEN      ********   X    01
HLD$GW_LNKCHN     ********   X    01
HLD$GW_SAVEFUNC   ********   X    01
HLD$NET_IO        ********   X    01
HLD$OPEN_TSKFIL   ********   X    01
HLD$PRINT         ********   X    01
HLD$PRTFAB        ********   X    01
HLD$PRTRAB        ********   X    01
HLD$RMS_CONNECT   ********   X    01
HLD$RMS_OPEN      ********   X    01
HLD$START         00000000 RG     01
IO$_READVBLK      ********   X    01
IO$_WRITEVBLK     ********   X    01
SS$_IVDEVNAM      ********   X    01
SYS$ASSIGN        ********  GX    01
SYS$EXIT          ********  GX    01
SYS$QIO           ********  GX    01
SYS$SETEF         ********  GX    01
SYS$TRNLOG        ********  GX    01
SYS$WAITFR        ********  GX    01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
HLD$CODE	00000100 (256.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.07	00:00:00.95
Command processing	132	00:00:00.59	00:00:04.94
Pass 1	128	00:00:01.37	00:00:09.38
Symbol table sort	0	00:00:00.02	00:00:00.01

Pass 2	58	00:00:00.58	00:00:02.22
Symbol table output	3	00:00:00.04	00:00:00.04
Psect synopsis output	1	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	361	00:00:02.70	00:00:17.57

The working set limit was 1050 pages.
4983 bytes (10 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 37 non-local and 7 local symbols.
227 source lines were read in Pass 1, producing 16 object records in Pass 2.
11 pages of virtual memory were used to define 11 macros.

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

Macro library name	Macros defined
_255DUA28:[HLD.OBJ]HLD.MLB;1	1
_255DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	11

99 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:HLDMAIN/OBJ=OBJ$:HLDMAIN MSRC$:HLDMAIN/UPDATE=(ENH$:HLDMAIN)+LIB$:HLD/LIB

0186 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INIT

HLDPRINT
LIS

IMGDMPDEF
SDL

INIT
MAP

HLDDATA
LIS

INIBAD
LIS

INIDEF
B32

HLDMAIN
LIS

ANALIMDMP
LIS

IMGDMP

ANALIMDMP
MAP

IMGDMP
MAP

HLDIO
LIS

DISPIMDMP
LIS

HLDFILE
LIS

HLDTASK
LIS

IMGDMP
LIS

INIALL
LIS