

HHH	HHH	LLL	DDDDDDDDDDDDDD	
HHH	HHH	LLL	DDDDDDDDDDDDDD	
HHH	HHH	LLL	DDDDDDDDDDDDDD	
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
HHHHHHHHHHHHHHHHHH	LLL	DDD	DDD	
HHHHHHHHHHHHHHHHHH	LLL	DDD	DDD	
HHHHHHHHHHHHHHHHHH	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	LLL	DDD	DDD
HHH	HHH	LLLLLLLLLLLLLLLL	DDDDDDDDDDDDDD	
HHH	HHH	LLLLLLLLLLLLLLLL	DDDDDDDDDDDDDD	
HHH	HHH	LLLLLLLLLLLLLLLL	DDDDDDDDDDDDDD	


```
HH      HH  LL      DDDDDDDD  FFFFFFFFFF  IIIIII  LL      EEEEEEEEE
HH      HH  LL      DDDDDDDD  FFFFFFFFFF  IIIIII  LL      EEEEEEEEE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DD        DD  FFFFFFFF  II      LL      EEEEEEE
HH      HH  LL      DD        DD  FFFFFFFF  II      LL      EEEEEEE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DD        DD  FF        II      LL      EE
HH      HH  LL      DDDDDDDD  FF        IIIIII  LLLLLLLLLL  EEEEEEEEE
HH      HH  LL      DDDDDDDD  FF        IIIIII  LLLLLLLLLL  EEEEEEEEE
```

```
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
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
```



```
0000 1      .TITLE HLD FILE -- HLD FILE I/O
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: 23-MAY-79
0000 43 :
0000 44 : MODIFICATIONS:
0000 45 :
0000 46 :     V001      SGD0002      19-Mar-1984
0000 47 :     --      Add error message for improper connect from SLD
0000 48 :
0000 49 :
0000 50 : INCLUDE FILES
0000 51 :
0000 52 :     $HLDDEF
0000 53 :
0000 54 : MACROS:
0000 55 :
0000 56 :     NONE
```



```

                                0000
                                0000
                                0000
                                00000000
                                0000
24 42 54 44 4C 48 00000008'010E0000' 0000
24 4B 53 41 54 48 00000016'010E0000' 000E
24 45 44 4F 4E 48 00000024'010E0000' 001C
    44 4E 45 2E 00000032'010E0000' 002A
    50 41 4D 0000003E'010E0000' 0036
    4D 4E 55 00000049'010E0000' 0041
    4E 55 4C 00000054'010E0000' 004C

58 :
59 : LOCAL DATA
60 :
61 : .PSECT HLD$PURE NOSHR,NOEXE,RD,NOWRT, LONG
62 :
63 FIRST_RECORD: .ASCID /HLDTB$/ ; THIS SHOULD MATCH THE 1ST REC IN HLD.DAT
64 TASK_FLAG: .ASCID /HTASK$/ ; FOR TASK ENTRIES
65 NODE_FLAG: .ASCID /HNODE$/ ; FOR NODE ENTRIES
66 END_FLAG: .ASCID /.END/ ; FOR END-OF-FILE
67 MAP_STRING: .ASCID /MAP/ ; FOR TASK TYPE MATCH
68 UNM_STRING: .ASCID /UNM/ ; DITTO
69 LUN_STRING: .ASCID /LUN/ ; DITTO

```



```
00000000 71      .PSECT  HLD$CODE          NOSHR,EXE,RD,NOWRT
0000      72
0000      73      ++
0000      74      FUNCTIONAL DESCRIPTION:
0000      75
0000      76      HLD$GET_FILE OPENS SYSS$SYSTEM:HLD.DAT AND SCANS IT FOR THE
0000      77      REQUIRED TASK.
0000      78
0000      79      INPUTS:
0000      80
0000      81      R1 - POINTS TO OPTIONAL DATA FIELD
0000      82
0000      83      OUTPUTS:
0000      84
0000      85      R11 - POINTS TO OPTDATA FIELD AFTER TASK NAME
0000      86      HLD$GW_SAVEFUNC - SAVES I/O FUNCTION FOR TRANSFER
0000      87
0000      88      --
0000      89
0000      90      .ENABLE LOCAL_BLOCK
0000      91
0000      92      HLD$GET_FILE::                      ; ENTRY POINT
0000      93
0000      94      VALIDATE THE CONNECT PARAMETERS
0000      95
0000      96      MOVB      (R1),R6                      ; Get no. of bytes of userdata
0000      97      MOVB      #3,(R1)+                  ; 3 bytes going back
0000      98      PUSHL     R1                          ; Save base of OPTDATA
0000      99      BSBW      HLD$C5TA                  ; DECODE THE TASK NAME
0000     100      POPL      R11                       ; Restore base of OPTDATA
0000     101      MOVAB     W^HLD$GT_INVALID,W^HLD$GT_OPER ; ASSUME INVALID REQUEST
0000     102      CMPB      B^HLD$B_CUN_FLAG(R11),#1 ; FIELD IN RANGE?
0000     103      BGTRU     90$                        ; IF GTRU NO
0000     104      MOVZBL    B^HLD$B_REQUEST(R11),R2 ; GET THE REQUEST
0000     105      CMPB      R6,#14                    ; RIGHT AMOUNT OF USERDATA?
0000     106      BNEQ      90$                        ; IF NEQ NO - INVALID CONNECT
0000     107      INCB      R2                        ; NORMALIZE
0000     108      BEQL      5$                        ; IF EQL OVERLAY REQUEST
0000     109      CMPB      R2,#3                    ; FIELD IN RANGE (-1 TO +2)?
0000     110      BGTRU     90$                        ; IF GTRU NO
0000     111      BNEQ      5$                        ; IF NEQ NOT CHECKPOINT WRITE
0000     112      MOVAB     W^HLD$NET_IO,W^HLD$GL_IOROUT 1 ; FIRST, GET NET DATA
0000     113      MOVAB     W^HLD$DISK_WRITE,W^HLD$GL_IOROUT 2 ; THEN DO DISK OPERATION
0000     114      MOVW      S^#IOS_READVBLK,W^HLD$GW_SAVEFUNC ; NETWORK DOES READ
0000     115      5$:
0000     116      MOVL      W^HLD$AT_REQ_TYPE[R2],W^HLD$GT_OPER ; SAVE PTR TO NAME OF REQUEST
0000     117
0000     118      ; NOW OPEN HLD.DAT AND SCAN IT FOR WHAT IS NEEDED
0000     119
0000     120      MOVAB     W^HLD$DATFAB,R0                ; SET TO OPEN FILE
0000     121      BSBW      HLD$RMS_OPEN                ; DO IT
0000     122      MOVAB     W^HLD$DATRAB,R0            ; SET TO CONNECT record STREAM
0000     123      BSBW      HLD$RMS_CONNECT            ; DO IT
0000     124
0000     125      ; PROCESS THE RECORDS
0000     126
0000     127      BSBW      GET_RECORD                    ; GET THE FIRST RECORD
```

56 61 90 0000 96
81 03 90 0003 97
51 DD 0006 98
FFF5' 30 0008 99
5B 8ED0 000B 100
0000'CF 0000'CF 9E 000E 101
01 00'AB 91 0015 102
67 1A 0019 103
52 00'AB 9A 001B 104
0E 56 91 001F 105
5E 12 0022 106
52 96 0024 107
1A 13 0026 108
03 52 91 0028 109
55 1A 002B 110
13 12 002D 111
0000'CF 0000'CF 9E 002F 112
0000'CF 0000'CF 9E 0036 113
0000'CF 00' B0 003D 114
0042 115
0000'CF 0000'CF42 D0 0042 116
004A 117
004A 118
004A 119
50 0000'CF 9E 004A 120
FFAE' 30 004F 121
50 0000'CF 9E 0052 122
FFA6' 30 0057 123
005A 124
005A 125
005A 126
0111 30 005A 127


```
58 0000'CF 9E 005D 128      MOVAB  W^FIRST RECORD,R8      ; SET TO MATCH
    0149 30 0062 129      BSBW   MATCH_STRING      ; IS STRING RIGHT?
    2D 50 E9 0065 130      BLBC   R0,105$           ; IF LBC NO
    0068 131      ;
    0068 132      ; SCAN FOR GENERAL PURPOSE TASKS
    0068 133      ;
    013C 30 0068 134      TRY_NEXT_TASK:
    1A 50 E9 006B 135      BSBW   FIND_TASK           ; HAVE A RECORD WITH 'HTASK$'?
    006E 136      BLBC   R0,100$           ; IF LBC NO - SCAN FOR NODE NAMES
    006E 137      ;
    006E 138      ; LOOK AT THE GENERAL PURPOSE TASKS FOR A POSSIBLE MATCH
    006E 139      ;
    015C 30 006E 140      BSBW   MATCH_TASK           ; TRY TO MATCH THE TASK NAME
    F4 50 E9 0071 141      BLBC   R0,TRY_NEXT_TASK    ; IF LBC NOT FOUND - LOOP
    00'AB 95 0074 142      TSTB   B^HLD$B_LUN_FLAG(R11) ; CAN SLD HANDLE GENERAL-PURPOSE TASKS?
    0F 13 0077 143      BEQL   100$             ; If EQL no - try for named nodes
    0000'CF 96 0079 144      INCB   W^HLD$GB_LUNFLAG    ; General purpose tasks require lun-fixing
    00'AB 95 007D 145      TSTB   B^HLD$B_REQUEST(R11) ; Is this a checkpoint operation?
    4C 15 0080 146      BLEQ   OPEN_TASK           ; If LEQ no - proceed
    0082 147      ;
    0082 148      ; ILLEGAL CONNECT REQUESTED
    0082 149      ;
    0082 150      90$:
    56 09 D0 0082 151      MOVL   #HLD$C_ERR_REQST,R6 ; Illegal request from SLD
    00DB 31 0085 152      BRW    COMMON_EXIT         ; Go to common code
    0088 153      ;
    0088 154      ; SCAN FOR NODE NAME MATCH
    0088 155      ;
    0088 156      100$:
    0000'CF 94 0088 157      CLRB   W^HLD$GB_GPFLAG     ; This is not a general purpose task
    00F1 30 008C 158      BSBW   RESCAN_RECORD        ; START AGAIN
    0134 30 008F 159      BSBW   FIND_NODE           ; FIND 'HNODE$'
    03 50 E8 0092 160      BLBS   R0,1T0$           ; IF LBS OK
    00BE 31 0095 161      105$: BRW    FORMAT_ERROR    ; FILE FORMAT ERROR
    0098 162      110$:
    00F5 30 0098 163      BSBW   NEXT_FIELD           ; GET TO THE NAME
    58 0000'CF 9E 009B 164      MOVAB W^HLD$GQ_NODEDESC,R8 ; POINT TO NODE DESCRIPTOR
    010B 30 00A0 165      BSBW   MATCH_STRING        ; SEE IF THERE IS A MATCH
    19 50 E8 00A3 166      BLBS   R0,150$           ; TRY FOR TASK NAME
    00A6 167      120$:
    00C5 30 00A6 168      BSBW   GET_RECORD           ; READ NEXT HLD.DAT RECORD
    011A 30 00A9 169      BSBW   FIND_NODE           ; SEARCH FOR A HNODE$ RECORD
    E9 50 E8 00AC 170      BLBS   R0,1T0$           ; IF LBS FOUND
    00CE 30 00AF 171      BSBW   RESCAN_RECORD        ; START AGAIN
    58 002A'CF 9E 00B2 172      MOVAB W^END_FLAG,R8    ; SEE IF AT END OF FILE
    00F4 30 00B7 173      BSBW   MATCH_STRING        ; MATCH WITH ".END"
    08 50 E8 00BA 174      BLBS   R0,160$           ; IF LBS MATCH - COULDN'T FIND MATCH
    E7 11 00BD 175      BRB     120$               ; TRY FOR HNODE$
    00BF 176      ;
    00BF 177      ; NODE FOUND - TRY FOR TASK MATCH
    00BF 178      ;
    00BF 179      150$:
    00E5 30 00BF 180      BSBW   FIND_TASK           ; LOOK FOR 'HTASK$'
    03 50 E8 00C2 181      BLBS   R0,170$           ; IF LBS PROCEED
    0098 31 00C5 182      160$: BRW    SEARCH_FAILURE ; COULDN'T FIND WHAT I WANTED
    00C8 183      170$:
    0102 30 00C8 184      BSBW   MATCH_TASK           ; IS THIS THE RIGHT TASK?
```



```
F1 50 E9 00CB 185 BLBC R0,150$ ; IF LBC NO - TRY FOR ANOTHER
      00CE 186
      00CE 187 .DISABLE LOCAL_BLOCK
      00CE 188
      00CE 189
      00CE 190 : TASK FOUND - GET READY TO OPEN UP THE FILE
      00CE 191
      00CE 192 OPEN_TASK:
      51 D6 00CE 193 INCL R1 ; MOVE PAST COMMA
      00BD 30 00D0 194 BSBW NEXT_FIELD ; DO IT
      3C 81 91 00D3 195 CMPB (R1)+,#^A/</ ; IS THIS THE RIGHT DELIMITER
      1D 12 00D6 196 BNEQ 20$ ; IF NEQ ERROR
      0000'CF 51 D0 00D8 197 MOVL R1,W^HLD$GL_TSKFNA ; PRESERVE PTR TO FILESPEC
      61 3F 3E 3A 00DD 198 LOCC #^A?>?,#63,(R1) ; FIND THE TERMINATOR
      78 13 00E1 199 BEQL SYNTAX_ERROR ; IF EQL NOT FOUND
      0000'CF 3F 50 83 00E3 200 SUBB3 R0,#63,W^HLD$GB_TSKFNS ; GET LTH OF FILESPEC
      51 D6 00E9 201 INCL R1 ; MOVE PAST '>'
      00A2 30 00EB 202 BSBW NEXT_FIELD ; FIND NEXT FIELD
      50 B5 00EE 203 TSTW R0 ; WAS THE RECORD EATEN?
      58 13 00F0 204 BEQL DONE ; If EQL yes
      2C 81 91 00F2 205 CMPB (R1)+,#^A",, ; NEXT DELIMITER OK?
      64 12 00F5 206 20$: BNEQ SYNTAX_ERROR ; IF NEQ NO
      0096 30 00F7 207 BSBW NEXT_FIELD ; MOVE TO NEXT FIELD
      58 0036'CF 9E 00FA 208 MOVAB W^MAP_STRING,R8 ; SET TO FIND FILE TYPE
      51 DD 00FF 209 PUSHL R1 ; SAVE THE POINTER FOR A LIMITED RESCAN
      00AA 30 0101 210 BSBW MATCH_STRING ; TRY MATCH
      12 50 E8 0104 211 BLBS R0,35$ ; IF LBS 'MAP'
      51 6E D0 0107 212 MOVL (SP),R1 ; RESTORE PTR
      58 0041'CF 9E 010A 213 MOVAB W^UNM_STRING,R8 ; TRY 'UNM'
      009C 30 010F 214 BSBW MATCH_STRING ; DO IT
      21 50 E9 0112 215 BLBC R0,80$ ; If LBC not 'UNM'
      0000'CF 96 0115 216 INCB W^HLD$GB_MAPFLAG ; SET 'UNM'
      0119 217 35$: TSTB W^HLD$GB_GPFLAG ; Is this a general-purpose task?
      0000'CF 95 0119 218 BEQL FORMAT_ERROR ; If EQL no - can't mix types
      37 13 011D 219 TSTB W^HLD$GB_MAPFLAG ; Mapped task?
      0000'CF 95 011F 220 BEQL 60$ ; If EQL yes
      07 13 0123 221 TSTW B^HLD$W_PART_ADDR(R11) ; Mapped task requested?
      00'AB B5 0125 222 BEQL 70$ ; If EQL yes, but this ain't it
      07 13 0128 223 BRB 100$ ; Finish in common code
      1C 11 012A 224 60$: TSTW B^HLD$W_PART_ADDR(R11) ; Mapped task requested?
      00'AB B5 012C 225 BEQL 100$ ; If EQL yes - proceed
      17 13 012F 226 70$: TSTL (SP)+ ; Clean up the stack
      8E D5 0131 227 BRW TRY_NEXT_TASK ; Keep looking for task
      FF32 31 0133 228 80$: MOVL (SP),R1 ; RESTORE PTR
      51 6E D0 0136 229 MOVAB W^LUN_STRING,R8 ; TRY 'LUN'
      58 004C'CF 9E 0139 230 BSBW MATCH_STRING ; MATCH?
      006D 30 013E 231 BLBC R0,SYNTAX_ERROR ; IF LBC NO
      17 50 E9 0141 232 INCB W^HLD$GB_LUNFLAG ; SET 'LUN'
      0000'CF 96 0144 233 100$: TSTL (SP)+ ; CLEAN UP STACK
      8E D5 0148 234 014A 235 014A 236 014A 237 014A 238 014A 239 014A 240 014A 241 DONE: $CLOSE FAB=W^HLD$DATFAB ; CLOSE THE DATA FILE
```


		05	0155	242	RSB		: DONE
			0156	243			
			0156	244	FORMAT_ERROR:		
56	01	D0	0156	245	MOVL	#HLD\$C_ERR_FMT,R6	: FORMAT ERROR
	08	11	0159	246	BRB	COMMON_EXIT	: GO TO COMMON EXIT
			015B	247			
			015B	248	SYNTAX_ERROR:		
56	02	D0	015B	249	MOVL	#HLD\$C_ERR_SYNTAX,R6	: SYNTAX ERROR
	03	11	015E	250	BRB	COMMON_EXIT	: GO TO COMMON CODE
			0160	251	SEARCH_FAILURE:		
56	03	D0	0160	252	MOVL	#HLD\$C_ERR_SERCH,R6	: FILE NOT FOUND
			0163	253	COMMON_EXIT:		
0000'CF	56	90	0163	254	MOVB	R6,W^HLD\$GB_ERRORFLG	: SAVE THE ERROR CODE
50	00'	D0	0168	255	MOVL	S^#SS\$NORMAL,R0	: No more messages in LOG
FE92'	31		016B	256	BRW	HLD\$EXIT_TO_VMS	


```
016E 258 :++
016E 259 :
016E 260 : GET_RECORD - THIS ROUTINE GETS A RECORD FROM THE OPEN FILE AND RETURNS
016E 261 : THE LENGTH AND ADDRESS OF THE RECORD.
016E 262 :
016E 263 : RESCAN_RECORD - THIS ROUTINE RESETS TO THE BEGINNING OF THE
016E 264 : CURRENT RECORD.
016E 265 :
016E 266 : NEXT_FIELD - THIS ROUTINE IGNORES BLANKS AND TABS
016E 267 :
016E 268 : INPUTS:
016E 269 :
016E 270 : HLD$DATRAB (GET_RECORD, RESCAN_RECORD)
016E 271 : R0,R1 - DESCRIPTOR OF REMAINDER OF RECORD (NEXT_FIELD)
016E 272 :
016E 273 : OUTPUTS:
016E 274 :
016E 275 : R0 - REMAINING CHARACTERS IN RECORD
016E 276 : R1 - POINTER TO FIRST NON-BLANK, NON-TAB CHARACTER IN RECORD
016E 277 :
016E 278 :--
016E 279 :
016E 280 GET_RECORD:
016E 281 $GET RAB=W^HLD$DATRAB ; GET THE FIRST RECORD
0179 282 CHECK_RMS ; SEE IF OK
51 D5 017C 283 TSTL R1 ; END-OF-FILE?
D6 13 017E 284 BEQL FORMAT_ERROR ; IF EQL YES - JUST GO AWAY
0180 285
0180 286 RESCAN_RECORD:
0180 287 MOVZWL W^HLD$GW_DATRSZ,R0 ; SET UP RECORD LENGTH
51 0000'CF 3C 0180 288 MOVL W^HLD$GL-DATRBF,R1 ; SET UP RECORD POINTER
0000'CF 51 50 C1 018A 289 ADDL3 R0,R1,W^HLD$GL_RECEND ; POINT TO END OF RECORD
0190 290 ; FALL THROUGH TO FIND 1ST
0190 291 ; INTERESTING CHARACTER
0190 292 NEXT_FIELD:
50 0000'CF 51 C3 0190 293 SUBL3 R1,W^HLD$GL_RECEND,R0 ; COMPUTE CHARS LEFT IN RECORD
0196 294 10$:
20 61 91 0196 295 CMPB (R1),#^X20 ; BLANK?
05 13 0199 296 BEQL 15$ ; IF EQL YES
09 61 91 019B 297 CMPB (R1),#^X9 ; TAB?
06 12 019E 298 BNEQ 20$ ; IF NEQ NO - DONE
51 D6 01A0 299 15$:
50 D7 01A2 300 INCL R1 ; MOVE THE POINTER
F0 12 01A4 301 DECL R0 ; ANOTHER CHARACTER EATEN
01A6 302 BNEQ 10$ ; IF NEQ MORE CHARS IN RECORD
05 01A6 303 20$:
304 RSB ; DONE
```



```

01A7 306 :++
01A7 307 :
01A7 308 : FIND_TASK - FIND A RECORD WITH A TASK DESCRIPTOR IN IT (HTASK$)
01A7 309 :
01A7 310 : OUTPUTS:
01A7 311 :     R0 - LBS=>SUCCESS, LBC=>FAILURE
01A7 312 :
01A7 313 : --
01A7 314 :
01A7 315 : FIND_TASK:
58 000E'CF 10 9E 01A7 316 :     BSBB GET_RECORD ; FETCH A RECORD
01A9 317 :     MOVAB W^TASK_FLAG,R8 ; SET TO MATCH 'HTASK$'
01AE 318 : ; FALL THROUGH TO TRY FOR MATCH AND RETURN
01AE 319 :
01AE 320 : ++
01AE 321 :
01AE 322 : MATCH_STRING - MATCH A STRING EXACTLY
01AE 323 :
01AE 324 : INPUTS:
01AE 325 :     R1 - POINTER TO ONE STRING
01AE 326 :     R8 - ADDRESS OF DESCRIPTOR OF STRING TO MATCH
01AE 327 :     HLD$GL_RECEND - FOR COMPUTING MAX NO. OF CHARS THAT CAN BE LEFT
01AE 328 :
01AE 329 : OUTPUTS:
01AE 330 :     R0 - LBS=>SUCCESS, LBC=>FAILURE
01AE 331 :
01AE 332 : --
01AE 333 :
01AE 334 : MATCH_STRING:
50 0000'CF 51 C3 01AE 335 :     SUBL3 R1,W^HLD$GL_RECEND,R0 ; COMPUTE MAX NO. OF CHARS LEFT
01B4 336 :     CMPH R0,(R8) ; ENOUGH TO MATCH?
01B7 337 :     BLSSU 100$ ; IF LSSU NO
01B9 338 :     CMPC3 (R8),(R1),a4(R8) ; MATCH?
01BE 339 :     BNEQ 100$ ; IF NEQ NO
01C0 340 :     INCL R0 ; SUCCESS
01C2 341 :     RSB ; DONE
01C3 342 : 100$:
01C3 343 :     CLRL R0 ; FAILURE
01C5 344 :     RSB ; DONE
01C6 345 :
01C6 346 :
01C6 347 : ++
01C6 348 :
01C6 349 : FIND_NODE - FIND A RECORD WITH A NODE DESCRIPTOR IN IT (HNODE$)
01C6 350 :
01C6 351 : OUTPUTS:
01C6 352 :     R0 - LBS=>SUCCESS, LBC=>FAILURE
01C6 353 :
01C6 354 : --
01C6 355 :
01C6 356 : FIND_NODE:
58 001C'CF 9E 11 01C6 357 :     MOVAB W^NODE_FLAG,R8 ; SET TO MATCH 'HNODE$'
01CB 358 :     BRB MATCH_STRING ; TRY FOR MATCH AND RETURN
01CD 359 :
01CD 360 : ++
01CD 361 :
01CD 362 : MATCH_TASK - THIS ROUTINE MATCHES THE REQUESTED NAME WITH ONE IN THE FILE

```



```

01CD 363 :
01CD 364 : OUTPUTS:
01CD 365 : R0 - LBS=>SUCCESS, LBC=>FAILURE
01CD 366 :
01CD 367 : --
01CD 368 :
01CD 369 MATCH_TASK:
61 07 C1 10 01CD 370 BSBB NEXT_FIELD : FIND START OF TASK NAME
51 DD 01CF 371 PUSH R1 : SAVE POINTER
2C 3A 01D1 372 LOCC #^A/,/,#7,(R1) : FIND THE DELIMITER
02 BA 01D5 373 POPR #^M<R1> : RESTORE POINTER
03 12 01D7 374 BNEQ 10$ : IF NEQ FOUND
FF7F 31 01D9 375 BRW SYNTAX_ERROR : SYNTAX ERROR
50 07 50 A3 01DC 376 10$: :
06 50 B1 01E0 377 SUBW3 R0,#7,R0 : COMPUTE LENGTH OF NAME IN RECORD
10 1A 01E3 378 CMPW R0,#6 : TOO BIG?
0001'CF 06 2D 01E5 379 BGTRU 100$ : IF GTRU YES
06 12 01ED 380 CMPC5 #6,W^HLD$AT_TSKBUF+1,#^X20,R0,(R1) : TRY TO MATCH TASK NAME
51 50 D6 01EF 381 BNEQ 100$ : IF NEQ FAILED
53 D0 01F1 382 INCL R0 : INDICATE SUCCESS
05 01F4 383 MOVL R3,R1 : RESTORE POINTER
01F5 384 RSB : DONE
50 D4 01F5 385 100$: :
05 01F7 386 CLRL R0 : INDICATE FAILURE
01F8 387 RSB : RETURN
01F8 388
01F8 389 .END

```


HLD FILE
Symbol table

- HLD FILE I/O

H 2

16-SEP-1984 01:40:52

5-SEP-1984 01:28:20

VAX/VMS Macro V04-00

[HLD.SRC]HLD FILE.MAR;1

Page 10

(1)

```

$$TMP1      = 00000001
$$TMP2      = 000000CF
COMMON_EXIT 00000163 R 03
DONE        0000014A R R 03
END_FLAG    0000002A R R 02
FIND_NODE   000001C6 R R 03
FIND_TASK   000001A7 R R 03
FIRST_RECORD 00000000 R R 02
FORMAT_ERROR 00000156 R R 03
GET_RECORD  0000016E R 03
HLD$AT_REQ_TYPE ***** X 03
HLD$AT_TSKBUF ***** X 03
HLD$B_CUN_FLAG ***** X 03
HLD$B_REQUEST ***** X 03
HLD$CSTA ***** X 03
HLD$CHECK_RMS ***** X 03
HLD$C_ERR_FMT= 00000001
HLD$C_ERR_REQST= 00000009
HLD$C_ERR_SERCH= 00000003
HLD$C_ERR_SYNTAX= 00000002
HLD$DATFAB ***** X 03
HLD$DATRAB ***** X 03
HLD$DISK_WRITE ***** X 03
HLD$EXIT_TO_VMS ***** X 03
HLD$GB_ERRORFLG ***** X 03
HLD$GB_GPFLAG ***** X 03
HLD$GB_LUNFLAG ***** X 03
HLD$GB_MAPFLAG ***** X 03
HLD$GB_TSKFNS ***** X 03
HLD$GET_FILE 00000000 RG 03
HLD$GL_DATRBF ***** X 03
HLD$GL_IOROUT_1 ***** X 03
HLD$GL_IOROUT_2 ***** X 03
HLD$GL_RECEND ***** X 03
HLD$GL_TSKFNA ***** X 03
HLD$GQ_NODEDESC ***** X 03
HLD$GT_INVALID ***** X 03
HLD$GT_OPER ***** X 03
HLD$GW_DATRSZ ***** X 03
HLD$GW_SAVEFUNC ***** X 03
HLD$NET_IO ***** X 03
HLD$RMS_CONNECT ***** X 03
HLD$RMS_OPEN ***** X 03
HLD$W_PART_ADDR ***** X 03
IOS_READVBLK ***** X 03
LUN_STRING  0000004C R 02
MAP_STRING  00000036 R 02
MATCH_STRING 000001AE R 03
MATCH_TASK  000001CD R 03
NEXT_FIELD   00000190 R 03
NODE_FLAG    0000001C R 02
OPEN_TASK    000000CE R 03
RESCAN_RECORD 00000180 R 03
SEARCH_FAILURE 00000160 R 03
SS$NORMAL ***** X 03
SYNTAX_ERROR 0000015B R 03
SYS$CLOSE ***** GX 03

```

```

SYSSGET ***** GX 03
TASK_FLAG 0000000E R 02
TRY_NEXT_TASK 00000068 R 03
UNM_STRING 00000041 R 02

```


+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR
HLD\$PURE	00000057 (87.)	02 (2.)	NOPIC USR
HLD\$CODE	000001F8 (504.)	03 (3.)	NOPIC USR

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	39	00:00:00.08	00:00:01.13
Command processing	149	00:00:00.63	00:00:03.29
Pass 1	145	00:00:01.85	00:00:06.00
Symbol table sort	0	00:00:00.06	00:00:00.18
Pass 2	81	00:00:00.89	00:00:03.43
Symbol table output	7	00:00:00.06	00:00:00.52
Psect synopsis output	2	00:00:00.03	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	426	00:00:03.60	00:00:14.58

The working set limit was 1200 pages.

8775 bytes (18 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 66 non-local and 21 local symbols.

389 source lines were read in Pass 1, producing 18 object records in Pass 2.

13 pages of virtual memory were used to define 11 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[HLD.OBJ]HLD.MLB;1	2
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	8

126 GETS were required to define 8 macros.

There were no errors, warnings or information messages.

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

0186

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY