


```
RRRRRRRR  DDDDDDDD  HH  HH  000000  MM  MM  EEEEEEEEE
RRRRRRRR  DDDDDDDD  HH  HH  000000  MM  MM  EEEEEEEEE
RR      RR  DD      DD  HH  HH  00      00  MMMM  MMMM  EE
RR      RR  DD      DD  HH  HH  00      00  MMMM  MMMM  EE
RR      RR  DD      DD  HH  HH  00      00  MM  MM  EE
RRRRRRRR  DD      DD  HHHHHHHHHH  00      00  MM  MM  EEEEEEE
RRRRRRRR  DD      DD  HHHHHHHHHH  00      00  MM  MM  EEEEEEE
RR      RR  DD      DD  HH  HH  00      00  MM  MM  EE
RR      RR  DD      DD  HH  HH  00      00  MM  MM  EE
RR      RR  DD      DD  HH  HH  00      00  MM  MM  EE
RR      RR  DDDDDDDD  HH  HH  000000  MM  MM  EEEEEEEEE
RR      RR  DDDDDDDD  HH  HH  000000  MM  MM  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
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

```
0001 0 MODULE RDHOME (  
0002 0     LANGUAGE (BLISS32),  
0003 0     IDENT = 'V04-000'  
0004 0 ) =  
0005 1 BEGIN  
0006 1  
0007 1 |  
0008 1 |*****  
0009 1 |*  
0010 1 |*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0011 1 |*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0012 1 |*  ALL RIGHTS RESERVED.  
0013 1 |*  
0014 1 |*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0015 1 |*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0016 1 |*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0017 1 |*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0018 1 |*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0019 1 |*  TRANSFERRED.  
0020 1 |*  
0021 1 |*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0022 1 |*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0023 1 |*  CORPORATION.  
0024 1 |*  
0025 1 |*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0026 1 |*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0027 1 |*  
0028 1 |*  
0029 1 |*****  
0030 1 |  
0031 1 |++  
0032 1 |  
0033 1 |FACILITY: MOUNT Utility Structure Levels 1 & 2  
0034 1 |  
0035 1 |ABSTRACT:  
0036 1 |    This routine reads the home block, if any, of the volume being mounted.  
0037 1 |  
0038 1 |ENVIRONMENT:  
0039 1 |  
0040 1 |    STARLET operating system, including privileged system services  
0041 1 |    and internal exec routines.  
0042 1 |  
0043 1 |--  
0044 1 |  
0045 1 |  
0046 1 |AUTHOR: Andrew C. Goldstein, CREATION DATE: 13-Oct-1977 21:29  
0047 1 |  
0048 1 |MODIFIED BY:  
0049 1 |  
0050 1 |  
0051 1 |    V03-003 HH0017      Hai Huang      3-May-1984  
0052 1 |    Use SURFACE_ERROR macro to determine if search for  
0053 1 |    the home block should continue.  
0054 1 |  
0055 1 |    V03-002 CDS0001    Christian D. Saether 13-Sep-1983  
0056 1 |    Do not set VALID. The drivers have been modified  
0057 1 |    to do this in conjunction with the on line count
```

RDHOME
V04-000

E 4
16-Sep-1984 01:31:05
14-Sep-1984 12:45:33

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MOUNT.SRC]RDHOME.B32;1 Page 2 (1)

```

: 58      0058 1 0 to 1 transition.
: 59      0059 1 Eliminate SET VALID routine and move CLEAR_VALID
: 60      0060 1 routine to VMOUNT as it is only referenced there
: 61      0061 1 on tape error paths.
: 62      0062 1
: 63      0063 1 V03-001 STJ0247 Steven T. Jeffreys, 31-Mar-1982
: 64      0064 1 Use a common I/O routine.
: 65      0065 1
: 66      0066 1 V02-004 STJ0002 Steven T. Jeffreys, 29-Aug-1980
: 67      0067 1 Liberal re-write to facilitate operator assisted mount.
: 68      0068 1 As a result, the code indentation is screwed up.
: 69      0069 1
: 70      0070 1 V02-003 ACG0167 Andrew C. Goldstein, 18-Apr-1980 13:39
: 71      0071 1 Previous revision history moved to MOUNT.REV
: 72      0072 1 !**
: 73      0073 1
: 74      0074 1
: 75      0075 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 76      0076 1 REQUIRE 'SRC$:MOUDEF.B32';
: 77      0608 1
: 78      0609 1
: 79      0610 1 FORWARD ROUTINE
: 80      0611 1 READ_HOMEBLOCK; ! read the home block
```

```
0612 1 GLOBAL ROUTINE READ_HOMEBLOCK (VOLUME_LABEL, FULL_SEARCH) =
0613 1
0614 1 ++
0615 1
0616 1 FUNCTIONAL DESCRIPTION:
0617 1
0618 1     This routine reads the home block, if any, of the volume being mounted.
0619 1     Various statuses are returned to be interpreted in the context of the
0620 1     operation; errors that are clear losers are signalled.
0621 1
0622 1
0623 1 CALLING SEQUENCE:
0624 1     READ_HOMEBLOCK (ARG1, ARG2)
0625 1
0626 1 INPUT PARAMETERS:
0627 1     ARG1: address of volume label string descriptor
0628 1     ARG2: 1 if search whole disk for home block
0629 1           0 if limited search
0630 1
0631 1 IMPLICIT INPUTS:
0632 1     CHANNEL: channel number assigned to device being mounted
0633 1
0634 1 OUTPUT PARAMETERS:
0635 1     NONE
0636 1
0637 1 IMPLICIT OUTPUTS:
0638 1     HOME_BLOCK: buffer contains home block if found
0639 1     HOMEBLOCK_LBN: LBN of home block read
0640 1
0641 1 ROUTINE VALUE:
0642 1     1 if valid and correct home block found
0643 1     $$$_NOHOMEBLK if home block not found
0644 1     $$$_INCVOLLABEL if home block found but wrong volume name
0645 1     $$$_DEVOFFLINE
0646 1     $$$_MEDOFL
0647 1     $$$_FILESTRUCT
0648 1
0649 1 SIDE EFFECTS:
0650 1     NONE
0651 1
0652 1 --
0653 1
0654 2 BEGIN
0655 2
0656 2 MAP
0657 2     VOLUME_LABEL      : REF VECTOR;      ! volume label string descriptor
0658 2
0659 2 LOCAL
0660 2     STATUS,           ! system service status
0661 2     IO_STATUS         : VECTOR[2],       ! I/O status block
0662 2     DELTA,            ! home block search delta
0663 2     BLOCKFACT,        ! device blocking factor
0664 2     LBN;              ! current LBN being tried
0665 2
0666 2 EXTERNAL
0667 2     CHANNEL,          ! channel number for I/O
0668 2     DEVICE_CHAR       : BBLOCK,          ! disk device characteristics
```

```
139 0669 2      DEVCHAR_DESC : VECTOR,      ! device characteristics descriptor
140 0670 2      HOME_BLOCK   : BBLOCK,      ! home block buffer
141 0671 2      HOME_BLOCK_LBN, ! LBN of home block read
142 0672 2      CTL$GL_PHD    : REF BBLOCK ADDRESSING_MODE (ABSOLUTE);
143 0673 2      ! pointer to process header
144 0674 2
145 0675 2  EXTERNAL ROUTINE
146 0676 2      READ_BLOCK,      ! read a disk block
147 0677 2      CHECK_HOME_BLK1, ! verify structure level 1 home block
148 0678 2      CHECK_HOME_BLK2; ! verify structure level 2 home block
149 0679 2
150 0680 2
151 0681 2
152 0682 2  ! Issue a pack acknowledge so we can talk to the disk.
153 0683 2  ! Inhibit error logging to avoid saturating the error log
154 0684 2  ! with media/device offline errors. These errors cause
155 0685 2  ! us to return and try an operator assisted mount.
156 0686 2
157 0687 2
158 P 0688 2  STATUS = DO_IO (CHAN = .CHANNEL,
159 P 0689 2      FUNC = (IO$PACKACK OR IO$M_INHERLOG),
160 P 0690 2      EFN = MOUNT_EFN,
161 0691 2      IOSB = IO_STATUS[0]);
162 0692 2  IF .STATUS THEN STATUS = -(IO_STATUS[0])<0,16>;
163 0693 2  IF NOT .STATUS
164 0694 2  THEN
165 0695 2      !
166 0696 2      ! Allow ILLIOFUNC errors, as devices that have
167 0697 2      ! no PACKACK function will return this status.
168 0698 2
169 0699 2      IF .STATUS NEQ SSS_ILLIOFUNC
170 0700 2      THEN
171 0701 2          ERR_EXIT (.STATUS);
172 0702 2
173 0703 2  ! Re-read the device characteristics. (For some devices, packack causes the
174 0704 2  ! correct characteristics to be determined and recorded in the I/O data base.)
175 0705 2
176 0706 2
177 0707 2  $GETCHN (CHAN = .CHANNEL, PRIBUF = DEVCHAR_DESC);
178 0708 2
179 0709 2  ! Compute the home block search delta from the volume geometry in the
180 0710 2  ! device table. This is done according to the following rules, where volume
181 0711 2  ! geometry is expressed in the order sectors, tracks, cylinders:
182 0712 2
183 0713 2      n x 1 x 1:      1
184 0714 2      1 x n x 1:      1
185 0715 2      1 x 1 x n:      1
186 0716 2
187 0717 2      n x m x 1:      n+1
188 0718 2      n x 1 x m:      n+1
189 0719 2      1 x n x m:      n+1
190 0720 2
191 0721 2      s x t x c:      (t+1)*s+1
192 0722 2
193 0723 2
194 0724 2  BLOCKFACT = (.DEVICE_CHAR[DIB$B_SECTORS]
195 0725 2      * .DEVICE_CHAR[DIB$B_TRACKS])
```

```
196 0726 3      * .DEVICE_CHAR[DIB$W_CYLINDERS])
197 0727 2      / .DEVICE_CHAR[DIB$L_MAXBLOCK];
198 0728 2
199 0729 2      DELTA = 1;
200 0730 2      IF .DEVICE_CHAR[DIB$W_CYLINDERS] GTR 1
201 0731 2      AND .DEVICE_CHAR[DIB$B_TRACKS] GTR 1
202 0732 2      THEN DELTA = .DELTA + .DEVICE_CHAR[DIB$B_TRACKS];
203 0733 2
204 0734 2      IF .DEVICE_CHAR[DIB$B_SECTORS] GTR 1
205 0735 2      AND (.DEVICE_CHAR[DIB$W_CYLINDERS] GTR 1
206 0736 2      OR .DEVICE_CHAR[DIB$B_TRACKS] GTR 1)
207 0737 2      THEN DELTA = (.DELTA * .DEVICE_CHAR[DIB$B_SECTORS] + .BLOCKFACT) / .BLOCKFACT;
208 0738 2
209 0739 2      IF .DELTA EQL 0
210 0740 2      OR .DELTA GTRU .DEVICE_CHAR[DIB$L_MAXBLOCK] / 10
211 0741 2      THEN DELTA = 1;
212 0742 2
213 0743 2      ! Limit the home block search to 10 hits to save time. If the device is
214 0744 2      being mounted Files-11 then we try all the way just in case the home block
215 0745 2      is in a weird location. Note the potential protection hole: Disks with the
216 0746 2      home block far into the disk are not protected from being mounted foreign
217 0747 2      by non-privileged users. C'est la vie.
218 0748 2
219 0749 2
220 0750 2      LBN = 1;
221 0751 2
222 0752 2      IF (
223 0753 3      DECRU J FROM (IF .FULL_SEARCH THEN -1 ELSE 10) TO 1 DO
224 0754 4      BEGIN
225 0755 4          STATUS = READ_BLOCK (.LBN, HOME_BLOCK);
226 0756 4          IF .STATUS
227 0757 4          THEN
228 0758 5              BEGIN
229 0759 5                  IF .HOME_BLOCK[HM2$B_STRUCLEV] EQL 1
230 0760 5                  THEN STATUS = CHECK_HOME_BLK1 (HOME_BLOCK, .LBN, .VOLUME_LABEL)
231 0761 5                  ELSE IF .HOME_BLOCK[HM2$B_STRUCLEV] EQL 2
232 0762 5                  THEN STATUS = CHECK_HOME_BLK2 (HOME_BLOCK, .LBN, .VOLUME_LABEL)
233 0763 5                  ELSE STATUS = 0;
234 0764 5                  IF .STATUS
235 0765 5                  OR .STATUS EQL SSS_INCVOLLABEL
236 0766 5                  THEN EXITLOOP 0;
237 0767 5                  END
238 0768 4          ELSE
239 0769 5              BEGIN
240 0770 5                  IF .STATUS EQL SSS_ILLBLKNUM
241 0771 5                  THEN EXITLOOP -1;
242 0772 6                  IF NOT SURFACE_ERROR (.STATUS)
243 0773 5                  THEN ERR_EXIT (.STATUS);
244 0774 4                  END;
245 0775 4
246 0776 4          IF NOT .FULL_SEARCH
247 0777 4          AND .BLOCK [CTL$GL PHD[PHD$Q_PRIVMSK], PRV$V_VOLPRO]
248 0778 4          THEN RETURN (SSS_NOHOMEBLK);
249 0779 4
250 0780 4          LBN = .LBN + .DELTA;
251 0781 4          END
252 0782 3      )
```

```
! end of routine READ_HOMEBLK
```

.TITLE	RDHOME	
.IDENT	\V04-000\	
.EXTRN	CHANNEL, DEVICE CHAR	
.EXTRN	DEVCHAR_DESC, HOME_BLOCK	
.EXTRN	HOMEBLOCK_LBN, CTL\$GL PHD	
.EXTRN	READ_BLOCK, CHECK HOME_BLK1	
.EXTRN	CHECK HOME_BLK2, COMMON_IO	
.EXTRN	SYSS\$GETCHN	
.PSECT	\$CODE\$,NOWRT,2	
.ENTRY	READ_HOMEBLOCK, Save R2,R3,R4,R5,R6,R7,R8,-	0612
MOVAB	R9	:
MOVAB	LIB\$STOP, R9	:
MOVAB	HOME_BLOCK, R8	:
MOVAB	DEVICE_CHAR+10, R7	:
SUBL2	#8, SP	:
CLRQ	-(SP)	: 0691
CLRQ	-(SP)	:
CLRQ	-(SP)	:
CLRQ	-(SP)	:
PUSHAB	IO STATUS	:
MOVZWL	#2056, -(SP)	:
PUSHL	CHANNEL	:
PUSHL	#26	:
CALLS	#12, COMMON_IO	:
MOVL	R0, STATUS	:
BLBC	STATUS, 1\$	: 0692
MOVZWL	IO STATUS, STATUS	:
BLBS	STATUS, 2\$	: 0693
CMPL	STATUS, #244	: 0699
BEQL	2\$	:
PUSHL	STATUS	: 0701
CALLS	#1, LIB\$STOP	:
CLRQ	-(SP)	: 0707
PUSHAB	DEVCHAR_DESC	:
CLRL	-(SP)	:
PUSHL	CHANNEL	:
CALLS	#5, SYSS\$GETCHN	:
MOVZBL	DEVICE_CHAR+8, R0	: 0724
MOVZBL	DEVICE_CHAR+9, R1	: 0725
MULL3	R1, R0, R2	:
MOVZWL	DEVICE_CHAR+10, R4	: 0726

54	52	66	54	C4	0006F	MULL2	R4, R2	:	0727
	52		A7	C7	00072	DIVL3	DEVICE_CHAR+112, R2, BLOCKFACT	:	0729
	52		01	D0	00077	MOVL	#1, DELTA	:	0730
	01		55	D4	0007A	CLRL	R5	:	
			67	B1	0007C	CMPW	DEVICE_CHAR+10, #1	:	
			0A	1B	0007F	BLEQU	3\$	:	
			55	D6	00081	INCL	R5	:	
	01		51	91	00083	CMPB	R1, #1	:	0731
			03	1B	00086	BLEQU	3\$	:	
	52		51	C0	00088	ADDL2	R1, DELTA	:	0732
	01		50	91	0008B	CMPB	R0, #1	:	0734
			12	1B	0008E	BLEQU	5\$	:	
	05		55	E8	00090	BLBS	R5, 4\$	:	0735
	01		51	91	00093	CMPB	R1, #1	:	0736
			0A	1B	00096	BLEQU	5\$	:	
	50		52	C4	00098	MULL2	DELTA, R0	:	0737
	50		54	C0	0009B	ADDL2	BLOCKFACT, R0	:	
52	50		54	C7	0009E	DIVL3	BLOCKFACT, R0, DELTA	:	
			52	D5	000A2	TSTL	DELTA	:	0739
			0A	13	000A4	BEQL	6\$	:	
50	66	A7	0A	C7	000A6	DIVL3	#10, DEVICE_CHAR+112, R0	:	0740
		50	52	D1	000AB	CMPL	DELTA, R0	:	
			03	1B	000AE	BLEQU	7\$	:	
	52		01	D0	000B0	MOVL	#1, DELTA	:	0741
	54		01	D0	000B3	MOVL	#1, LBN	:	0750
	05	08	AC	E9	000B6	BLBC	FULL_SEARCH, 8\$	:	0753
	55		01	CE	000BA	MNEGL	#1, R5	:	
			03	11	000BD	BRB	9\$	:	
	55		0A	D0	000BF	MOVL	#10, R5	:	
	56	08	AC	D2	000C2	MCOML	FULL_SEARCH, R6	:	0776
			008F	31	000C6	BRW	18\$	:	0780
		0110	8F	BB	000C9	PUSHR	#*M<R4,R8>	:	0755
	0000G	CF	02	FB	000CD	CALLS	#2, READ_BLOCK	:	
		53	50	D0	000D2	MOVL	R0, STATUS	:	
		3B	53	E9	000D5	BLBC	STATUS, 15\$	:	0756
		01	A8	91	000D8	CMPB	HOME_BLOCK+13, #1	:	0759
			0E	12	000DC	BNEQ	11\$	:	
		04	AC	DD	000DE	PUSHL	VOLUME_LABEL	:	0760
			54	DD	000E1	PUSHL	LBN	:	
	0000G	CF	58	DD	000E3	PUSHL	R8	:	
			03	FB	000E5	CALLS	#3, CHECK_HOME_BLK1	:	
			12	11	000EA	BRB	12\$	:	
	02	0D	A8	91	000EC	CMPB	HOME_BLOCK+13, #2	:	0761
			11	12	000F0	BNEQ	13\$	:	
		04	AC	DD	000F2	PUSHL	VOLUME_LABEL	:	0762
			54	DD	000F5	PUSHL	LBN	:	
			58	DD	000F7	PUSHL	R8	:	
	0000G	CF	03	FB	000F9	CALLS	#3, CHECK_HOME_BLK2	:	
		53	50	D0	000FE	MOVL	R0, STATUS	:	
			02	11	00101	BRB	14\$	:	
			53	D4	00103	CLRL	STATUS	:	0763
	0000010C	66	53	E8	00105	BLBS	STATUS, 21\$	:	0764
		8F	53	D1	00108	CMPL	STATUS, #268	:	0765
			34	12	0010F	BNEQ	16\$	:	
			5B	11	00111	BRB	21\$	:	0766
	000000DC	8F	53	D1	00113	CMPL	STATUS, #220	:	0770
			43	13	0011A	BEQL	19\$	:	

000001F4	8F	53	D1	0011C	CMPL	STATUS, #500	: 0772
		20	13	00123	BEQL	16\$	:
0000005C	8F	53	D1	00125	CMPL	STATUS, #92	:
		17	13	0012C	BEQL	16\$	:
000000BC	8F	53	D1	0012E	CMPL	STATUS, #188	:
		0E	13	00135	BEQL	16\$	:
00002144	8F	53	D1	00137	CMPL	STATUS, #8516	:
		05	13	0013E	BEQL	16\$	:
		53	DD	00140	PUSHL	STATUS	: 0773
	69	01	FB	00142	CALLS	#1, LIB\$STOP	:
	0B	56	E9	00145	BLBC	R6, 17\$	: 0776
15	50	9F	D0	00148	MOVL	@#CTL\$GL PHD, R0	: 0777
	60	15	E0	0014F	BBS	#21, (R0), 20\$	:
	54	52	C0	00153	ADDL2	DELTA, LBN	: 0780
		55	D7	00156	DECL	J	: 0753
		55	D5	00158	TSTL	J	:
		03	13	0015A	BEQL	19\$	:
		FF6A	31	0015C	BRW	10\$	:
		58	DD	0015F	PUSHL	R8	: 0785
		01	DD	00161	PUSHL	#1	:
0000G	CF	02	FB	00163	CALLS	#2, READ_BLOCK	:
	50	8F	3C	00168	MOVZWL	#2272, R0	: 0786
			04	0016D	RET		:
0000G	CF	54	D0	0016E	MOVL	LBN, HOMEBLOCK_LBN	: 0789
	50	53	D0	00173	MOVL	STATUS, R0	: 0791
			04	00176	RET		: 0793

; Routine Size: 375 bytes, Routine Base: \$CODE\$ + 0000

: 264 0794 1
: 265 0795 1 END
: 266 0796 0 ELUDOM

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	375	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	24	0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:RDHOME/OBJ=OBJ\$:RDHOME MSRC\$:RDHOME/UPDATE=(ENH\$:RDHOME)

: Size: 375 code + 0 data bytes
: Run Time: 00:14.5
: Elapsed Time: 00:33.3
: Lines/CPU Min: 3298
: Lexemes/CPU-Min: 29954
: Memory Used: 154 pages
: Compilation Complete

0246 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

