

MMM	MMM	TTTTTTTTTTTTTTTT	AAAAAAAAAA	AAAAAAAAAA	CCCCCCCCCCCC	PPPPPPPPPPPP	
MMM	MMM	TTTTTTTTTTTTTTTT	AAAAAAAAAA	AAAAAAAAAA	CCCCCCCCCCCC	PPPPPPPPPPPP	
MMM	MMM	TTTTTTTTTTTTTTTT	AAAAAAAAAA	AAAAAAAAAA	CCCCCCCCCCCC	PPPPPPPPPPPP	
MMMMMM	MMMMMM	TTT	AAA	AAA	CCC	PPP	PPP
MMMMMM	MMMMMM	TTT	AAA	AAA	CCC	PPP	PPP
MMMMMM	MMMMMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM	MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM	MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAA	AAA	CCCCCCCCCCCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAA	AAA	CCCCCCCCCCCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAA	AAA	CCCCCCCCCCCC	PPPPPPPPPPPP	

```
RRRRRRRR      WW      WW      VV      VV      000000      LL
RRRRRRRR      WW      WW      VV      VV      000000      LL
RR      RR      WW      WW      VV      VV      00      00      LL
RR      RR      WW      WW      VV      VV      00      00      LL
RR      RR      WW      WW      VV      VV      00      00      LL
RR      RR      WW      WW      VV      VV      00      00      LL
RRRRRRRR      WW      WW      VV      VV      00      00      LL
RRRRRRRR      WW      WW      VV      VV      00      00      LL
RR      RR      WW      WW      VV      VV      00      00      LL
RR      RR      WW      WW      VV      VV      00      00      LL
RR      RR      WWW      WWW      VV      VV      00      00      LL
RR      RR      WWW      WWW      VV      VV      00      00      LL
RR      RR      WW      WW      VV      VV      000000      LLLLLLLLLL
RR      RR      WW      WW      VV      VV      000000      LLLLLLLLLL
```

```
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
0002 0 MODULE RWVOL (LANGUAGE (BLISS32),
0003 0 IDENT = 'V04-000',
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1
0032 1 FACILITY: MTAACP
0033 1
0034 1 ABSTRACT:
0035 1 This modules rewinds a volume set
0036 1
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: D. H. GILLESPIE, CREATION DATE: 16-JUL-1977
0047 1
0048 1 MODIFIED BY:
0049 1
0050 1 V03-002 MMD0002 Meg Dumont, 3-Jan-1983 16:15
0051 1 Add modifier IOSM_CLRSEEXCP to all QIO's issued by the MTAACP,
0052 1 necessary for the MSCP tape drives.
0053 1
0054 1 V03-001 MMD0001 Meg Dumont, 5-Nov-1982 16:42
0055 1 Changed SET_VALID to a QIO IOS_PACKACK.
0056 1
0057 1 V02-007 DMW00070 David Michael Walp 20-Jan-1981
```

```

58 0058 1 | Added Extra Set Valid in REWIND_AND_WAIT
59 0059 1 |
60 0060 1 | V02-006 DMW00015 David Michael Walp 14-Mar-1981
61 0061 1 | Changed calculation of CCB address to GET_CCB
62 0062 1 |
63 0063 1 | V02-005 REFORMAT Maria del C. Nasr 30-Jun-1980
64 0064 1 |
65 0065 1 | **
66 0066 1 |
67 0067 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
68 0068 1 |
69 0069 1 | REQUIRE 'SRC$:MTADEF.B32';
70 0453 1 |
71 0454 1 | FORWARD ROUTINE
72 0455 1 | REWIND_AND_WAIT : COMMON_CALL, | rewind and wait for
73 0456 1 | | completion
74 0457 1 | REQUEST_UNBLOCK : COMMON_CALL NOVALUE, | AST to handles completion of
75 0458 1 | | rewind
76 0459 1 | REWIND_VOL_SET : COMMON_CALL NOVALUE, | rewind volume set
77 0460 1 | SET_AMB_POS : COMMON_CALL NOVALUE, | set the position of vol set
78 0461 1 | | to ambiguous
79 0462 1 | SET_VALID : COMMON_CALL NOVALUE; | set valid bit
80 0463 1 |
81 0464 1 | EXTERNAL
82 0465 1 | CURRENT_UCB : REF BBLOCK, | address of current unit
83 0466 1 | | control block
84 0467 1 | CURRENT_WCB : REF BBLOCK, | address of current window
85 0468 1 | | control block
86 0469 1 | IO_CHANNEL, | ACP IO channel
87 0470 1 | IO_PACKET, | address of current IO packet
88 0471 1 | IO_STATUS, | IO status
89 0472 1 | USER_STATUS : VECTOR [2]; | user IO status
90 0473 1 |
91 0474 1 | EXTERNAL ROUTINE
92 0475 1 | BLOCK, | block current request
93 0476 1 | CLOSE_FILE : L$CLOSE_FILE, | close a file
94 0477 1 | DO_CANCEL : COMMON_CALL, | cancel all IO
95 0478 1 | SYSS$QIO : ADDRESSING_MODE (ABSOLUTE), | QIO function
96 0479 1 | UNBLOCK; | unblock a request
97 0480 1 |
```



```

99      0481 1 GLOBAL ROUTINE REWIND_VOL_SET : COMMON_CALL NOVALUE =
100     0482 1
101     0483 1 ++
102     0484 1
103     0485 1 FUNCTIONAL DESCRIPTION:
104     0486 1     This routine rewinds the volumes in a volume set
105     0487 1     waits for the completion of the rewind of volume one
106     0488 1     and sets the current volume set position to ambiguous
107     0489 1
108     0490 1 CALLING SEQUENCE:
109     0491 1     REWIND_VOL_SET()
110     0492 1
111     0493 1 INPUT PARAMETERS:
112     0494 1     none
113     0495 1
114     0496 1 IMPLICIT INPUTS:
115     0497 1     CURRENT_VCB - address of current volume control block
116     0498 1     CURRENT_WCB - address of current window control block
117     0499 1
118     0500 1 OUTPUT PARAMETERS:
119     0501 1     none
120     0502 1
121     0503 1 IMPLICIT OUTPUTS:
122     0504 1     volumes rewound
123     0505 1     current relative volume number and channel set to first volume
124     0506 1
125     0507 1 ROUTINE VALUE:
126     0508 1     none
127     0509 1
128     0510 1 SIDE EFFECTS:
129     0511 1     SS$_FILALRACC - can not rewind volume set if a file is currently accessed
130     0512 1     SS$_FCPREWINDERR - file control process rewind error
131     0513 1
132     0514 1 --
133     0515 1
134     0516 2 BEGIN
135     0517 2
136     0518 2 EXTERNAL REGISTER
137     0519 2     COMMON_REG;
138     0520 2
139     0521 2 EXTERNAL ROUTINE
140     0522 2     GET_CCB;
141     0523 2
142     0524 2 LOCAL
143     0525 2     MVL      : REF BBLOCK,
144     0526 2     NVOLS,
145     0527 2     VOLLIST : REF BBLOCKVECTOR [, MVL$K LENGTH],
146     0528 2     CCB      : REF BBLOCK,
147     0529 2     UCBLIST : REF VECTOR;
148     0530 2
149     0531 2 ! first make sure that there are no files accessed
150     0532 2 !
151     0533 2
152     0534 2 IF .CURRENT_WCB NEQU 0
153     0535 2 THEN
154     0536 2     ERR_EXIT(SS$_FILALRACC);
155     0537 2
```

```
156 0538 2 IF .CURRENT_VCB[VCBSV_PARTFILE]
157 0539 2 THEN
158 0540 2 CLOSE_FILE();
159 0541 2
160 0542 2 ! setup pointers to interesting structures
161 0543 2
162 0544 2 MVL = .CURRENT_VCB[VCBSL_MVL];
163 0545 2 NVOLS = .MVL[MVL$B_NVOLS] - 1;
164 0546 2 VOLLIST = .MVL + MVL$K_FIXLEN;
165 0547 2 UCBLIST = BBLOCK[.CURRENT_VCB[VCBSL_RVT], RVT$K_UCBLST];
166 0548 2 CCB = KERNEL_CALL (GET_CCB, .IO_CHANNEL);
167 0549 2 KERNEL_CALL (SET_AMB_POS);
168 0550 2
169 0551 2 ! now issue rewinds
170 0552 2
171 0553 2
172 0554 2 DECR I FROM .NVOLS TO 0 DO
173 0555 2 BEGIN
174 0556 2
175 0557 2 IF .VOLLIST[I, MVL$V_MOUNTED]
176 0558 2 THEN
177 0559 2 BEGIN
178 0560 2
179 0561 2 ! assign channel
180 0562 2
181 0563 2 CCB[CCBSL_UCB] = .UCBLIST[VOLLIST[I, MVL$B_RVN]];
182 0564 2
183 0565 2 IF .I NEQU 0 ! don't wait for all but first
184 0566 2 THEN
185 P 0567 2 $QIOW(CHAN = .IO_CHANNEL,
186 0568 2 FUNC = IO$REWIND OR IO$NOWAIT OR IO$CLSEREXCP)
187 0569 2 ELSE
188 0570 2 REWIND_AND_WAIT();
189 0571 2
190 0572 2 END;
191 0573 2
192 0574 2 END;
193 0575 2
194 0576 1 END;
```

```
.TITLE RWVOL
.IDENT \V04-000\
```

```
.EXTRN CURRENT_UCB, CURRENT_WCB
.EXTRN IO_CHANNEL, IO_PACKET
.EXTRN IO_STATUS, USER_STATUS
.EXTRN BLOCK, CLOSE_FILE
.EXTRN DO_CANCEL, SYSSQIO
.EXTRN UNBLOCK, GET_CCB
.EXTRN SYSSCMKRN, SYSSQIOW
```

```
.PSECT $CODE$,NOWRT,2
```

```
56 00000000G 047C 00000
0000G 9F 9E 00002
CF D5 00009
```

```
.ENTRY REWIND_VOL_SET, Save R2,R3,R4,R5,R6,R10 : 0481
MOVAB @#SYSSCMKRN, R6 :
TSTL CURRENT_WCB : 0534
```


			04	13	0000D	BEQL	1\$		
			8F	BF	0000F	CHMU	#164		0536
03		00A4	AB	E9	00013	BLBC	11(CURRENT_VCB), 2\$		0538
		0B				BSBW	CLOSE FILE		0540
50		34	AB	D0	00017	MOVL	52(CURRENT_VCB), MVL		0544
52		0B	A0	9A	0001E	MOVZBL	11(MVL), NVOLS		0545
			52	D7	00022	DECL	NVOLS		
53		24	A0	9E	00024	MOVAB	36(R0), VOLLIST		0546
54	20	AB	00000044	8F	C1	ADDL3	#68, 32(CURRENT_VCB), UCBLIST		0547
			0000G	CF	DD	PUSHL	IO_CHANNEL		0548
				01	DD	PUSHL	#1		
				5E	DD	PUSHL	SP		
				CF	9F	PUSHAB	GET_CCB		
66			0000G	04	FB	CALLS	#4, -SYS\$CMKRNL		
55				50	D0	MOVL	R0, CCB		
				7E	D4	CLRL	-(SP)		0549
				5E	DD	PUSHL	SP		
				CF	9F	PUSHAB	SET_AMB_POS		
66			0000V	03	FB	CALLS	#3, -SYS\$CMKRNL		
				52	D6	INCL	I		0557
				3A	11	BRB	5\$		
32				07	A342	PUSHAQ	7(VOLLIST)[I]		
				00	E1	BBC	#0, a(SP)+ 5\$		
				06	A342	PUSHAQ	6(VOLLIST)[I]		0563
				9E	9A	MOVZBL	a(SP)+ R0		
50				6440	D0	MOVL	(UCBLIST)[R0], (CCB)		
65				52	D5	TSTL	I		0565
				1E	13	BEQL	4\$		
				7E	7C	CLRQ	-(SP)		0568
				7E	7C	CLRQ	-(SP)		
				7E	7C	CLRQ	-(SP)		
				7E	7C	CLRQ	-(SP)		
				7E	7C	CLRQ	-(SP)		
				7E	D4	CLRL	-(SP)		
7E		02A4	8F	3C	00073	MOVZWL	#676, -(SP)		
		0000G	CF	DD	00078	PUSHL	IO_CHANNEL		
			7E	D4	0007C	CLRL	-(SP)		
000C0000G	00		0C	FB	0007E	CALLS	#12, SYS\$QIOW		
			05	11	00085	BRB	5\$		
0000V	CF		00	FB	00087	CALLS	#0, REWIND_AND_WAIT		0570
	C3		52	F4	0008C	SOBGEQ	I, 3\$		0554
			04	0008F	RET				0576

; Routine Size: 144 bytes, Routine Base: \$CODE\$ + 0000

; 195 0577 1

```
197 0578 1 GLOBAL ROUTINE SET_AMB_POS : COMMON_CALL NOVALUE =
198 0579 1
199 0580 1 ++
200 0581 1
201 0582 1 FUNCTIONAL DESCRIPTION:
202 0583 1 This routine set the position of the volume set to ambiguous
203 0584 1
204 0585 1 CALLING SEQUENCE:
205 0586 1 SET_AMB_POS()
206 0587 1 call in kernel mode
207 0588 1
208 0589 1 INPUT PARAMETERS:
209 0590 1 none
210 0591 1
211 0592 1 IMPLICIT INPUTS:
212 0593 1 CURRENT_VCB - address of current volume control block
213 0594 1
214 0595 1 OUTPUT PARAMETERS:
215 0596 1 none
216 0597 1
217 0598 1 IMPLICIT OUTPUTS:
218 0599 1 none
219 0600 1
220 0601 1 ROUTINE VALUE:
221 0602 1 none
222 0603 1
223 0604 1 SIDE EFFECTS:
224 0605 1 none
225 0606 1
226 0607 1 --
227 0608 1
228 0609 1 BEGIN
229 0610 2
230 0611 2 EXTERNAL REGISTER
231 0612 2 COMMON_REG;
232 0613 2
233 0614 2 CURRENT_VCB[VCB$B_TM] = 0;
234 0615 2 CURRENT_VCB[VCB$B_ST_RECORD] = 0;
235 0616 2 CURRENT_VCB[VCB$B_LOGICEOVS] = 0;
236 0617 2 CURRENT_VCB[VCB$B_CUR_RVN] = 0;
237 0618 2 CURRENT_VCB[VCB$B_CUR_FID] = 0;
238 0619 1 END;
```

```
0000 00000
2E AB 94 00002
30 AB D4 00005
OB AB 02 8A 00008
2F AB 94 0000C
24 AB D4 0000F
04 00012
```

```
.ENTRY SET AMB POS, Save nothing
CLRB 46(CURRENT_VCB)
CLRL 48(CURRENT_VCB)
BICB2 #2, 11(CURRENT_VCB)
CLRB 47(CURRENT_VCB)
CLRL 36(CURRENT_VCB)
RET
```

```
: 0578
: 0614
: 0615
: 0616
: 0617
: 0618
: 0619
```

; Routine Size: 19 bytes, Routine Base: \$CODE\$ + 0090

RWVOL
V04-000

⁹
16-Sep-1984 02:32:38
14-Sep-1984 12:46:50

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[MTAACP.SRC]RWVOL.B32;1⁷ (3)

: 239

0620 1

STR
V04

```
241 0621 1 GLOBAL ROUTINE REQUEST_UNBLOCK (VCB) : COMMON_CALL NOVALUE =
242 0622 1
243 0623 1 ++
244 0624 1
245 0625 1 FUNCTIONAL DESCRIPTION:
246 0626 1 Handles requests to unblock current volume operation
247 0627 1
248 0628 1 CALLING SEQUENCE:
249 0629 1 REWIND_DONE(ARG1)
250 0630 1
251 0631 1 INPUT PARAMETERS:
252 0632 1 ARG1 - address of VCB
253 0633 1
254 0634 1 IMPLICIT INPUTS:
255 0635 1 saved stack and impure area
256 0636 1
257 0637 1 OUTPUT PARAMETERS:
258 0638 1 none
259 0639 1
260 0640 1 IMPLICIT OUTPUTS:
261 0641 1 none
262 0642 1
263 0643 1 ROUTINE VALUE:
264 0644 1 none
265 0645 1
266 0646 1 SIDE EFFECTS:
267 0647 1 never returns to PC where AST's were enabled
268 0648 1 instead it resumes where the blocked request left off
269 0649 1
270 0650 1 --
271 0651 1
272 0652 2 BEGIN
273 0653 2
274 0654 2 EXTERNAL REGISTER
275 0655 2 COMMON_REG;
276 0656 2
277 0657 2 CURRENT_VCB = .VCB; ! restore VCB
278 0658 2
279 0659 2 IF .CURRENT_VCB[VCB$V_CANCELIO]
280 0660 2 THEN
281 0661 3 BEGIN
282 0662 3 ERROR(SS$ CANCEL);
283 0663 3 KERNEL_CALL(DO_CANCEL); ! cancel IO
284 0664 3 IO_PACKET = 0; ! no request packet to process
285 0665 3 RETURN;
286 0666 3
287 0667 2 END;
288 0668 2
289 0669 2 UNBLOCK(); ! unblock and continue where request processing left off
290 0670 1 END;
```

5B 04 AC 0000 0000
DO 00002

ENTRY REQUEST_UNBLOCK, Save nothing
MOVL VCB, CURRENT_VCB

: 0621
: 0657

RWVOL
V04-000

N 9
16-Sep-1984 02:32:38
14-Sep-1984 12:46:50

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]RWVOL.B32;1

Page 9
(4)

1B	OB	AB		05	E1	00006	BBC	#5, 11(CURRENT VCB), 1\$	:	0659
	0000G	CF	0830	8F	B0	0000B	MOVW	#2096, USER_STATUS	:	0662
				7E	D4	00012	CLRL	-(SP)	:	0663
				5E	DD	00014	PUSHL	SP	:	
			0000G	CF	9F	00016	PUSHAB	DO_CANCEL	:	
	00000000G	9F		03	FB	0001A	CALLS	#3, @#SYSS\$CMKRNL	:	
			0000G	CF	D4	00021	CLRL	IO_PACKET	:	0664
					04	00025	RET		:	0661
	0000G	CF		00	FB	00026	CALLS	#0, UNBLOCK	:	0669
				04	0002B	1\$:	RET		:	0670

; Routine Size: 44 bytes, Routine Base: \$CODE\$ + 00A3

; 291 0671 1

```
0672 1 GLOBAL ROUTINE REWIND_AND_WAIT : COMMON_CALL =
0673 1
0674 1 !++
0675 1
0676 1 FUNCTIONAL DESCRIPTION:
0677 1     This routine rewinds the unit assigned to IO_CHANNEL
0678 1     and waits for completion of the rewind
0679 1
0680 1 CALLING SEQUENCE:
0681 1     REWIND_AND_WAIT()
0682 1
0683 1 INPUT PARAMETERS:
0684 1     none
0685 1
0686 1 IMPLICIT INPUTS:
0687 1     CURRENT_VCB - address of current volume control block
0688 1     IO_CHANNEL - ACP IO channel
0689 1
0690 1 OUTPUT PARAMETERS:
0691 1     none
0692 1
0693 1 IMPLICIT OUTPUTS:
0694 1     none
0695 1
0696 1 ROUTINE VALUE:
0697 1     0 - unit off_line
0698 1     1 - successful
0699 1
0700 1 SIDE EFFECTS:
0701 1     none
0702 1
0703 1 --
0704 1
0705 2 BEGIN
0706 2
0707 2 EXTERNAL REGISTER
0708 2     COMMON_REG;
0709 2
0710 2 LOCAL
0711 2     STATUS;                                ! status of IO
0712 2
0713 2     STATUS = $QIOW( CHAN = .IO_CHANNEL,
0714 2                     FUNC = IOS$PACKACK OR IOSM_CLSEREXCP,
0715 2                     IOSB = IO STATUS);
0716 2     SYSSQIO(0, .IO_CHANNEL, IOS$REWIND OR IOSM_CLSEREXCP,
0717 2             BBLOCK[CURRENT_VCB[VCBSL_VPFL],
0718 2             VVP$L_STATUS], REQUEST_UNBLOCK, .CURRENT_VCB, 0,0,0,0,0,0);
0719 2     BLOCK($FIELDMASK(VCBS$V_WAIREWIND));
0720 2     STATUS = .BBLOCK[CURRENT_VCB[VCBSL_VPFL], VVP$L_STATUS];
0721 2
0722 2 IF .STATUS<0, 16> EQL SSS$MEDOFL
0723 2 OR
0724 2     .STATUS<0, 16> EQL SSS$VOLINV
0725 2 THEN
0726 2     RETURN 0;
0727 2
0728 2 IF NOT .STATUS
```


350
351
352
353
354
355
356
357
358
359
360
361
362

0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741

2
2
2
2
2
2
2
2
2
2
2
2
1

```
THEN
  BEGIN
    USER_STATUS[0] = STATUS;
    USER_STATUS[1] = $$$_FCPREWNDERR;
    ERR_EXIT();
  END;

  STATUS = $QIOW( CHAN = IO_CHANNEL,
                  FUNC = IOS_PACKACK OR IOSM_CLSEREXCP,
                  IOSB = IO_STATUS);

  RETURN 1;

END;
```

54	0000G	CF	9E	00002	.ENTRY	REWIND AND WAIT, Save R2,R3,R4	0672
53	00000000G	00	9E	00007	MOVAB	IO_CHANNEL, R4	
		7E	7C	0000E	MOVAB	SY\$\$QIOW, R3	
		7E	7C	00010	CLRQ	-(SP)	0715
		7E	7C	00012	CLRQ	-(SP)	
		7E	7C	00014	CLRQ	-(SP)	
	0000G	CF	9F	00016	CLRQ	-(SP)	
7E	0208	8F	3C	0001A	PUSHAB	IO_STATUS	
		64	DD	0001F	MOVZWL	#520, -(SP)	
		7E	D4	00021	PUSHL	IO_CHANNEL	
63		0C	FB	00023	CLRL	-(SP)	
52		50	D0	00026	CALLS	#12, SY\$\$QIOW	
		7E	7C	00029	MOVL	R0, STATUS	0718
		7E	7C	0002B	CLRQ	-(SP)	
		7E	7C	0002D	CLRQ	-(SP)	
		5B	DD	0002F	CLRQ	-(SP)	
	A0	AF	9F	00031	PUSHL	CURRENT_VCB	
7E	3C	8F	C1	00034	PUSHAB	REQUEST_UNBLOCK	
		8F	3C	0003D	ADDL3	#412, 60(CURRENT_VCB), -(SP)	
		64	DD	00042	MOVZWL	#548, -(SP)	
		7E	D4	00044	PUSHL	IO_CHANNEL	
00000000G	9F	0C	FB	00046	CLRL	-(SP)	
		08	DD	0004D	CALLS	#12, a#SY\$\$QIO	0719
0000G	CF	01	FB	0004F	PUSHL	#8	
	50	AB	D0	00054	CALLS	#1, BLOCK	
	52	C0	D0	00058	MOVL	60(CURRENT_VCB), R0	0720
01A4	8F	52	B1	0005D	MOVL	412(R0), STATUS	
		37	13	00062	CMPW	STATUS, #420	0722
0254	8F	52	B1	00064	BEQL	2\$	
		30	13	00069	CMPW	STATUS, #596	0724
	0E	52	E8	0006B	BEQL	2\$	
0000G	CF	52	D0	0006E	BLBS	STATUS, 1\$	0728
0000G	CF	8F	3C	00073	MOVL	STATUS, USER_STATUS	0731
		00	BF	0007A	MOVZWL	#2192, USER_STATUS+4	0732
		7E	7C	0007C	CHMU	#0	0733
		7E	7C	0007E	CLRQ	-(SP)	0738
		7E	7C	00080	CLRQ	-(SP)	
		7E	7C	00082	CLRQ	-(SP)	

RWVOL
V04-000

D 10
16-Sep-1984 02:32:38
14-Sep-1984 12:46:50

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]RWVOL.B32;1

Page 12
(5)

7E	0000G	CF	9F	00084	PUSHAB	IO STATUS	:
	0208	8F	3C	00088	MOVZWL	#520, -(SP)	:
		64	DD	0008D	PUSHL	IO CHANNEL	:
		7E	D4	0008F	CLRL	-(SP)	:
63		0C	FB	00091	CALLS	#12, SYSSQIOW	:
52		50	D0	00094	MOVL	R0, STATUS	:
50		01	D0	00097	MOVL	#1, R0	: 0739
			04	0009A	RET		:
		50	D4	0009B 2\$:	CLRL	R0	: 0741
			04	0009D	RET		:

; Routine Size: 158 bytes, Routine Base: \$CODE\$ + 00CF

; 363 0742 1


```
0743 1 ROUTINE SET_VALID : COMMON_CALL NOVALUE =
0744 1
0745 1 ++
0746 1
0747 1 FUNCTIONAL DESCRIPTION:
0748 1     This routine issues the PACKACK on a volume.
0749 1
0750 1 CALLING SEQUENCE:
0751 1     SET_VALID()
0752 1
0753 1 INPUT PARAMETERS:
0754 1     none
0755 1
0756 1 IMPLICIT INPUTS:
0757 1     CURRENT_VCB - address of current volume control block
0758 1
0759 1 OUTPUT PARAMETERS:
0760 1     none
0761 1
0762 1 IMPLICIT OUTPUTS:
0763 1     VALID bit set
0764 1
0765 1 ROUTINE VALUE:
0766 1     none
0767 1
0768 1 SIDE EFFECTS:
0769 1     none
0770 1
0771 1 --
0772 1
0773 2 BEGIN
0774 2
0775 2     LOCAL
0776 2     STATUS ;
0777 2
0778 2     STATUS = $QIOW( CHAN = .IO_CHANNEL,
0779 2                     FUNC = IOS$PACKACK OR IOSM_CLSEREXCP,
0780 2                     IOSB = IO_STATUS);
0781 1 END;
```

0000 0000 SET_VALID:

				WORD	Save nothing	
7E	7C	00002		CLRQ	-(SP)	0743
7E	7C	00004		CLRQ	-(SP)	0780
7E	7C	00006		CLRQ	-(SP)	
7E	7C	00008		CLRQ	-(SP)	
	0000G	CF	9F	0000A	PUSHAB	IO_STATUS
	7E	0208	8F	3C	0000E	MOVZWL #520, -(SP)
	0000G	CF	DD	00013	PUSHL	IO_CHANNEL
		7E	D4	00017	CLRL	-(SP)
00000000G	00	0C	FB	00019	CALLS	#12, SYS\$QIOW
			04	00020	RET	0781

RWVOL
V04-000

F 10
16-Sep-1984 02:32:38
14-Sep-1984 12:46:50

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]RWVOL.B32;1 Page 14
(6)

; Routine Size: 33 bytes, Routine Base: \$CODE\$ + 0160

; 404 0782 1 END
; 405 0783 1
; 406 0784 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	398	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	35	0	1000	00:01.9

COMMAND QUALIFIERS

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

; Size: 398 code + 0 data bytes
; Run Time: 00:12.4
; Elapsed Time: 00:25.2
; Lines/CPU Min: 3781
; Lexemes/CPU-Min: 22543
; Memory Used: 110 pages
; Compilation Complete

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY