


```

RRRRRRRR MM MM 222222 FFFFFFFF MM MM TTTTTTTTTT BBBB8888 KK KK TTTTTTTTTT
RRRRRRRR MM MM 222222 FFFFFFFF MM MM TTTTTTTTTT BBBB8888 KK KK TTTTTTTTTT
RR RR RR MMMM MMMM 22 22 FF FF MMMM MMMM TT TT BB BB KK KK TT
RR RR RR MMMM MMMM 22 22 FF FF MMMM MMMM TT TT BB BB KK KK TT
RR RR RR MM MM MM 22 22 FF FF MM MM TT TT BB BB KK KK TT
RR RR RR MM MM MM 22 22 FF FF MM MM TT TT BB BB KK KK TT
RRRRRRRR MM MM 22 FFFFFFFF MM MM TT BBBB8888 KKKKKK TT
RRRRRRRR MM MM 22 FFFFFFFF MM MM TT BBBB8888 KKKKKK TT
RR RR MM MM 22 FF MM MM TT BB BB KK KK TT
RR RR MM MM 22 FF MM MM TT BB BB KK KK TT
RR RR MM MM 22 FF MM MM TT BB BB KK KK TT
RR RR MM MM 22 FF MM MM TT BB BB KK KK TT
RR RR MM MM 2222222222 FF MM MM TT BBBB8888 KK KK TT
RR RR MM MM 2222222222 FF MM MM TT BBBB8888 KK KK TT

```

```

LL LL 111111 SSSSSSSS
LL LL 111111 SSSSSSSS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SSSSSS
LL LL 11 SSSSSS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS

```

(2) 71
(3) 95

DECLARATIONS
RMSFMT_BKT2 - RELATIVE FILE BUCKET FORMAT ROUTINE

```

0000 1          $BEGIN RM2FMTBKT,000,RMSRMS2,<RELATIVE-SPECIFIC EXTEND>
0000 2
0000 3 :*****
0000 4 :*
0000 5 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 6 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 7 :*  ALL RIGHTS RESERVED.
0000 8 :*
0000 9 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 10 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 11 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 12 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 13 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 14 :*  TRANSFERRED.
0000 15 :*
0000 16 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 17 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 18 :*  CORPORATION.
0000 19 :*
0000 20 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 21 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 22 :*
0000 23 :*
0000 24 :*****
0000 25
0000 26 :++
0000 27 : FACILITY: RMS32
0000 28
0000 29 : ABSTRACT:
0000 30 :   Routine to perform RELATIVE file organization specific
0000 31 :   EXTEND processing, that is, the formatting of the data buckets
0000 32 :   by writing them with zeroes.
0000 33
0000 34 : ENVIRONMENT:
0000 35 :   STAR processor running STARLET EXEC.
0000 36
0000 37 : AUTHOR: L F Laverdure,          CREATION DATE: 1-Dec-1977
0000 38
0000 39 : MODIFIED BY:
0000 40
0000 41 :   V03-003 KBT0133          Keith B. Thompson          20-Aug-1982
0000 42 :   Reorganize psects
0000 43
0000 44 :   V03-002 KBT0115          Keith B. Thompson          6-Aug-1982
0000 45 :   Remove ref. to set_sifb_ptr
0000 46
0000 47 :   V03-001 KBT0096          Keith B. Thompson          13-Jul-1982
0000 48 :   Clean up psects
0000 49
0000 50 :   V02-007 CDS0002          C Saether                9-Jan-1982
0000 51 :   Fix code that depends on exact page size buffer.
0000 52
0000 53 :   V02-006 CDS00001        C. Saether                28-Aug-1981
0000 54 :   Don't actually lock EBK when doing formatting, only
0000 55 :   get a buffer.
0000 56
0000 57 :   V02-005 PSK00001        Paulina S. Knibbe         24-Jul-1981

```

0000	58	:	
0000	59	:	Fix some broken branches caused by integrating
0000	60	:	Prologue three changes.
0000	61	:	
0000	62	:	V02-004 REFORMAT Frederick E. Deen, Jr. 28-Jul-1980
0000	63	:	This code was reformatted to adhere to RMS standards
0000	64	:	
0000	65	:	V003 RAN0003 R A Newell 9-Nov-1978
0000	66	:	File sharing code enhancements
0000	67	--	
0000	68	:	
0000	69	:	

```
0000 71      .SBTTL  DECLARATIONS
0000 72
0000 73 :
0000 74 : INCLUDE FILES:
0000 75 :
0000 76 :
0000 77 :
0000 78 : MACROS:
0000 79 :
0000 80
0000 81      $BDBDEF
0000 82      $CSHDEF
0000 83      $IFBDEF
0000 84      $RMSDEF
0000 85
0000 86 :
0000 87 : EQUATED SYMBOLS:
0000 88 :
0000 89 :
0000 90 :
0000 91 : OWN STORAGE:
0000 92 :
0000 93
```

```
0000 95 .SBTTL RMSFMT_BKT2 - RELATIVE FILE BUCKET FORMAT ROUTINE
0000 96
0000 97 :++
0000 98 : RMSFMT_BKT2 - Relative File bucket format routine
0000 99 :
0000 100 : This module causes the blocks of a new extent of a RELATIVE file
0000 101 : to be written with zeroes.
0000 102 :
0000 103 : CALLING SEQUENCE:
0000 104 :
0000 105 :     BSBW    RMSFMT_BKT2
0000 106 :
0000 107 : INPUT PARAMETERS:
0000 108 :
0000 109 :     R11    IMPURE AREA address
0000 110 :     R10    IFAB address
0000 111 :     R9     IRAB/IFAB address
0000 112 :     R8     RAB/FAB address
0000 113 :     R6     END VBN of EXTENT +1
0000 114 :     R1     First VBN of EXTENT
0000 115 :
0000 116 : IMPLICIT INPUTS:
0000 117 :
0000 118 :     It is assumed that the PROLOG is locked.
0000 119 :     If IFB$L_EBK is less than R1, zeroing will start with VBN IFB$L_EBK.
0000 120 :
0000 121 : OUTPUT PARAMETERS:
0000 122 :
0000 123 :     R0          STATUS
0000 124 :     R1-R5,AP   Destroyed
0000 125 :
0000 126 : IMPLICIT OUTPUTS:
0000 127 :
0000 128 :     None
0000 129 :
0000 130 : COMPLETION CODES:
0000 131 :
0000 132 :     Standard RMS.
0000 133 :
0000 134 : SIDE EFFECTS:
0000 135 :
0000 136 :     None
0000 137 :
0000 138 : --
0000 139
```

```

0000 141
0000 142
0000 143 : Get a buffer to zero the EXTENT.
0000 144 : NOTE: By asking for a 1-byte buffer will get minimum of 1 block
0000 145 : (will actually get an entire bucket).
0000 146
0000 147
0000 148 RMSFMT_BKT2::
51 74 AA D1 0000 149 CMPL IFB$$_EBK(R10),R1 ; EOF < start of extent?
0004 150 BGEQU 5$ ; branch if not
51 74 AA D0 0006 151 MOVL IFB$$_EBK(R10),R1 ; yes - start zeroing with EBK
000A 152 5$: PUSHL R1 ; save starting VBN on stack.
000C 153 $CACHE VBN=#0,SIZE=#1,- ; get a buffer for formatting.
000C 154 FLAGS=<LOCK,NOREAD>,ERR=ERREX
001A 155 POPL BDB$$_VBN(R4) ; store start VBN in BDB
50 16 A4 3C 001E 156 MOVZWL BDB$$_SIZE(R4),R0 ; get buffer length
50 01FF 8F AA 0022 157 BICW2 #^X1FF, R0 ; only use full pages
7E 50 17 9C 0027 158 ROTL #23,R0,-(SP) ; get # of blocks in buffer
14 A4 50 B0 002B 159 MOVW R0,BDB$$_NUMB(R4) ; use entire buffer for zeroing
65 50 00 6E 00 2C 0031 160 PUSHL R4 ; save BDB address
0037 161 MOVCS #0,(SP),#0,R0,(R5) ; zero the buffer
0039 162 POPR #^M<R4,R5> ; restore BDB addr & # blocks
0039 163
0039 164 :
0039 165 : Loop writing a buffer-full of zeroes each time thru.
0039 166 :
0039 167
7E 55 1C A4 C1 0039 168 10$: ADDL3 BDB$$_VBN(R4),R5,-(SP) ; compute END VBN + 1 of blocks
003E 169 ; to be zeroed this write
56 6E D1 003E 170 CMPL (SP),R6 ; past end of extent?
0041 171 BLEQU 20$ ; branch if not
50 56 1C A4 C3 0043 172 SUBL3 BDB$$_VBN(R4),R6,R0 ; compute # blks for last xfer
14 A4 0200 8F 50 A5 0048 173 MULW3 R0,#5T2,BDB$$_NUMB(R4) ; and make into # bytes
004F 174 20$: BSBW RM$SEQWTBD ; go write the blocks
1C A4 8E D0 0052 175 MOVL (SP)+,BDB$$_VBN(R4) ; set start VBN for next xfer
11 50 E9 0056 176 BLBC R0,ERRXFR ; branch if write failed
56 1C A4 D1 0059 177 CMPL BDB$$_VBN(R4),R6 ; all done?
DA 1F 005D 178 BLSSU 10$ ; branch if not
005F 179
005F 180 :
005F 181 : All set, release the buffer and return to caller.
005F 182 :
005F 183
1C A4 D4 005F 184 RLS: CLRL BDB$$_VBN(R4) ; Zero VBN field again.
FF9B' 30 0062 185 BSBW RM$RLNERR ; Release buffer.
05 0065 186 RSB ; Return.
0066 187
5E 04 C0 0066 188 ERREX: ADDL2 #4,SP ; Clean start VBN off stack.
05 0069 189 RSB ; And return.
006A 190
006A 191 :
006A 192 : An error occurred on the WRITE to the file.
006A 193 : Release the buffer and return.
006A 194 :
006A 195
006A 196 ERRXFR:
50 DD 006A 197 PUSHL R0 ; Save error code.

```


F1	10	006C	198	BSBB	RLS		
01	BA	006E	199	POPR	#^M<R0>		; Clean up, release the buffer.
	05	0070	200	RSB			; restore status
		0071	201				
		0071	202				
		0071	203				
				.END			

```

$$PSECT_EP      = 00000000
$$TMP           = 00000005
$$RMSTEST      = 0000001A
$$RMS_PBUGCHK  = 00000010
$$RMS_TBUGCHK  = 00000008
$$RMS_UMODE    = 00000004
BDB$$_VBN      = 0000001C
BDB$$_NUMB     = 00000014
BDB$$_SIZE     = 00000016
CSH$$_LOCK     = 00000001
CSH$$_NOBUFFER = 00000008
CSH$$_NOREAD   = 00000004
ERREX         = 00000066 R    01
ERRXFR        = 0000006A R    01
IFB$$_EBK     = 00000074
RLS           = 0000005F R    01
RM$$_CACHE    = ***** X    01
RM$$_FMT_BKT2 = 00000000 RG   01
RM$$_RLNERR   = ***** X    01
RM$$_SEQWTBD  = ***** X    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
RM\$RMS2	00000071 (113.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.06	00:00:01.39
Command processing	110	00:00:00.64	00:00:04.98
Pass 1	214	00:00:04.65	00:00:15.12
Symbol table sort	0	00:00:00.49	00:00:01.00
Pass 2	51	00:00:00.96	00:00:02.26
Symbol table output	5	00:00:00.05	00:00:00.07
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	414	00:00:06.88	00:00:24.90

The working set limit was 1200 pages.
 24557 bytes (48 pages) of virtual memory were used to buffer the intermediate code.
 There were 30 pages of symbol table space allocated to hold 462 non-local and 3 local symbols.
 203 source lines were read in Pass 1, producing 13 object records in Pass 2.
 15 pages of virtual memory were used to define 14 macros.

Macro library statistics !

Macro library name	Macros defined
-----	-----
-\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	7
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	10

568 GETS were required to define 10 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RM2FMTBKT/OBJ=OBJ\$:RM2FMTBKT MSRC\$:RM2FMTBKT/UPDATE=(ENH\$:RM2FMTBKT)+EXECML\$/LIB+LIB\$:RMS/LIB

RM2CREATE LIS	RM2GET LIS	RM2PUT LIS	RM2EXTEND LIS	RM2MTBKT LIS	RM2OPEN LIS	RM2UPDEL LIS	RM3ALLBKT LIS	RM3BKTIO LIS	RM3BKT SPL LIS	RM3CLOSE LIS	RM3CMPKEY LIS	RM3CMPRSS LIS	RM3BUG LIS
---------------	------------	------------	---------------	--------------	-------------	--------------	---------------	--------------	----------------	--------------	---------------	---------------	------------