


```

RRRRRRRR MM MM 000000 CCCCCCCC HH HH KK KK SSSSSSSS UU UU MM MM
RRRRRRRR MM MM 000000 CCCCCCCC HH HH KK KK SSSSSSSS UU UU MM MM
RR RR RR MMMM MMMM 00 00 CC CCCCCC HH HH KK KK SS UU UU MMMM MMMM
RR RR RR MMMM MMMM 00 00 CC CCCCCC HH HH KK KK SS UU UU MMMM MMMM
RR RR RR MM MM 00 0000 CC MM MM HH HH KK KK SS UU UU MM MM MM
RRRRRRRR MM MM 00 00 0000 CC MM MM HH HH KK KK SS SSSSSS UU UU MM MM
RRRRRRRR MM MM 00 00 00 CC MM MM HH HH KK KK SS SSSSSS UU UU MM MM
RR RR MM MM 0000 00 CC CC HH HH KK KK SS UU UU MM MM
RR RR MM MM 0000 00 CC CC HH HH KK KK SS UU UU MM MM
RR RR MM MM 00 00 CC CC HH HH KK KK SS UU UU MM MM
RR RR MM MM 00 00 CC CC HH HH KK KK SS UU UU MM MM
RR RR MM MM 000000 CCCCCCCC HH HH KK KK SSSSSSSS UUUUUUUUUU MM MM
RR RR MM MM 000000 CCCCCCCC HH HH KK KK SSSSSSSS UUUUUUUUUU MM MM

```

```

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
LLLLLLLL IIIIII SSSSSSSS
LLLLLLLL IIIIII SSSSSSSS

```

(2)	58
(3)	79

DECLARATIONS
RMSCHKSUM - CALCULATE AND CHECK OR STORE CHECKSUM ROUTINES

```
0000 1          $BEGIN RMOCHKSUM,000,RMSRMS3,<PROLOG CHECKSUM ROUTINES>-  
0000 2                                     <PIC,NOSHR,NOWRT>  
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 : Facility: rms32  
0000 30 :  
0000 31 : Abstract:  
0000 32 :  
0000 33 :   this module provides two routines to handle the checksum  
0000 34 :   word for file prologs.  
0000 35 :  
0000 36 : Environment:  
0000 37 :   star processor running starlet exec.  
0000 38 :  
0000 39 : Author: L F Laverdure,      Creation Date: 10-OCT-1977  
0000 40 :  
0000 41 : Modified By:  
0000 42 :  
0000 43 :   V03-002 KBT0201      Keith B. Thompson      23-Aug-1982  
0000 44 :   Reorganize psects  
0000 45 :  
0000 46 :   V03-001 KBT0101      Keith B. Thompson      13-Jul-1982  
0000 47 :   Clean up psects  
0000 48 :  
0000 49 :   V02-004 MCN0001      Maria del C. Nasr      10-Jun-1981  
0000 50 :   Change PSECT to fix broken branch to this routine.  
0000 51 :  
0000 52 :   V02-003 REFORMAT      D M WALP      24-JUL-1980  
0000 53 :  
0000 54 :--  
0000 55 :  
0000 56 :
```

```
0000 58          .SBTTL  DECLARATIONS
0000 59
0000 60 :
0000 61 : Include Files:
0000 62 :
0000 63 :
0000 64 :
0000 65 : Macros:
0000 66 :
0000 67 :
0000 68          $RMSDEF
0000 69
0000 70 :
0000 71 : Equated Symbols:
0000 72 :
0000 73 :
0000 74 :
0000 75 : Own Storage:
0000 76 :
0000 77
```

```
0000 79 .SBTTL RM$CHKSUM - CALCULATE AND CHECK OR STORE CHECKSUM ROUTINES
0000 80
0000 81 :++
0000 82 : RM$CHKSUM
0000 83 :
0000 84 : entry at rm$chksum - calculate checksum and compare with stored
0000 85 : checksum.
0000 86 : entry at rm$maksum - calculate checksum and store.
0000 87 :
0000 88 : Calling sequence:
0000 89 :
0000 90 :     bsbw  rm$chksum
0000 91 :     or   bsbw  rm$maksum
0000 92 :
0000 93 : Input Parameters:
0000 94 :
0000 95 :     r5 = buffer addr
0000 96 :
0000 97 : Implicit Inputs:
0000 98 :
0000 99 :     none
0000 100 :
0000 101 : Output Parameters:
0000 102 :
0000 103 :     r0    status code
0000 104 :     r1,r2 destroyed
0000 105 :
0000 106 : Implicit Outputs:
0000 107 :
0000 108 :     none
0000 109 :
0000 110 : Completion Codes:
0000 111 :
0000 112 :     success=-1, failure=rm$_plg
0000 113 :
0000 114 : Side Effects:
0000 115 :
0000 116 :     none
0000 117 :
0000 118 : --
0000 119
```

```

0000 121
0000 122 :++
0000 123 : entry point to check the checksum
0000 124 :--
0000 125
0000 126 RMSCHKSUM::
52 0C 10 0000 127 RSB  CALSUM ; calculate checksum
61 B1 0002 128 CMPW (R1),R2 ; is stored checksum same
0005 129 ; as calculated value?
17 12 0005 130 BNEQ ERRPLG ; branch if not
05 0007 131 RSB ; return
0008 132
0008 133 :++
0008 134 : entry point to store the checksum
0008 135 :--
0008 136
0008 137 RMSMAKSUM::
51 04 10 0008 138 BSBB CALSUM ; calculate sum
52 B0 000A 139 MOVW R2,(R1) ; store it
05 000D 140 RSB ; and return
000E 141
000E 142 :++
000E 143 : calculate check sum routine
000E 144 :
000E 145 : input: r5 = buffer addr
000E 146 :
000E 147 : output: r0 = -1
000E 148 : r1 = addr of checksum word in buffer
000E 149 : r2 = calculate checksum value
000E 150 :--
000E 151
000E 152
000E 153 CALSUM:
50 52 D4 000E 154 CLRL R2 ; init checksum
51 55 D0 0010 155 MOVL R5,R1 ; copy buffer addr
FE 8F 9A 0013 156 MOVZBL #254.,R0 ; # words to sum -1
52 81 A0 0017 157 10$: ADDW2 (R1)+,R2 ; sum it up
FA 50 F4 001A 158 SOBGEQ R0,10$ ; loop 255 times
05 001D 159 RSB
001E 160
001E 161 :++
001E 162 : handle error
001E 163 :--
001E 164
001E 165 ERRPLG:
05 001E 166 RMSERR PLG ; declare error
0023 167 RSB
0024 168 .END

```

RMOCHKSUM
Symbol table

PROLOG CHECKSUM ROUTINES

L 1

16-SEP-1984 00:13:46
5-SEP-1984 16:21:28

VAX/VMS Macro V04-00
[RMS.SRC]RMOCHKSUM.MAR;1

Page 5
(5)

```

$$PSECT_EP      = 00000000
$$RMSTEST       = 0000001A
$$RMS_PBUGCHK   = 00000010
$$RMS_TBUGCHK   = 00000008
$$RMS_UMODE     = 00000004
CALSUM          = 0000000E R    01
ERRPLG          = 0000001E R    01
RMSCHKSUM       = 00000000 RG   01
RMSMAKSUM       = 00000008 RG   01
RMS$PLG        = 0001861C

```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR I.OEXE NORD NOWRT NOVEC BYTE
RMSRMS3	00000024 (36.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
\$ABSS	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	00:00:00.09	00:00:00.63
Command processing	134	00:00:00.69	00:00:04.21
Pass 1	171	00:00:02.55	00:00:10.03
Symbol table sort	0	00:00:00.14	00:00:00.19
Pass 2	41	00:00:00.61	00:00:02.11
Symbol table output	3	00:00:00.02	00:00:00.17
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	383	00:00:04.13	00:00:17.38

The working set limit was 1200 pages.
 12248 bytes (24 pages) of virtual memory were used to buffer the intermediate code.
 There were 20 pages of symbol table space allocated to hold 229 non-local and 1 local symbols.
 168 source lines were read in Pass 1, producing 13 object records in Pass 2.
 12 pages of virtual memory were used to define 11 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	4
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	7

311 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

RMOCHKSUM
VAX-11 Macro Run Statistics

PROLOG CHECKSUM ROUTINES

M 1

16-SEP-1984 00:13:46 VAX/VMS Macro V04-00
5-SEP-1984 16:21:28 [RMS.SRC]RMOCHKSUM.MAR;1

Page 6
(5)

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

