


```

DDDDDDDD  UU      UC  MM      MM  PPPPPPP
DDDDDDDD  UU      UU  MM      MM  PPPPPPP
DD      DD  UU      UU  MMMM  MMMM  PP      PP
DD      DD  UU      UU  MMMM  MMMM  PP      PP
DD      DD  UU      UU  MM  MM  MM  PP      PP
DD      DD  UU      UU  MM  MM  MM  PPPPPPP
DD      DD  UU      UU  MM  MM  MM  PPPPPPP
DD      DD  UU      UU  MM  MM  MM  PP
DD      DD  UU      UU  MM  MM  MM  PP
DD      DD  UU      UU  MM  MM  MM  PP
DD      DD  UU      UU  MM  MM  MM  PP
DD      DD  UU      UU  MM  MM  MM  PP
DDDDDDDD  UUUUUUUUU  MM      MM  PP
DDDDDDDD  UUUUUUUUU  MM      MM  PP

```

```

....
....
....
....

```

```

LL      IIIIII  SSSSSSS
LL      IIIIII  SSSSSSS
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  SSSSSSS
LLLLLLLL  IIIIII  SSSSSSS

```

(1)	2	COPYRIGHT NOTICE
(1)	29	PROGRAM DESCRIPTION
(2)	56	DECLARATIONS
(3)	65	STORAGE DEFINITIONS
(4)	76	SHOW_SYSTEM - DUMP SYSTEM REGION VIRTUAL MEMORY
(5)	115	SHOW_PO -- SHOW PO VIRTUAL MEMORY
(6)	148	SHOW_P1 -- DISPLAY P1 VIRTUAL MEMORY
(7)	190	DUMP - DUMP SPECIFIED MEMORY AREA

```

0000 1 .TITLE DUMP MEMORY DUMP ROUTINES
0000 2 .SBTTL COPYRIGHT NOTICE
0000 3 .IDENT 'V04-000'
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 29 .SBTTL PROGRAM DESCRIPTION
0000 30 :++
0000 31 : FACILITY
0000 32 : SYSTEM DUMP ANALYZER
0000 33 :
0000 34 : ABSTRACT
0000 35 :
0000 36 : DUMP MEMORY ROUTINES
0000 37 :
0000 38 : ENVIRONMENT
0000 39 :
0000 40 : NATIVE MODE, USER MODE
0000 41 :
0000 42 : AUTHOR
0000 43 :
0000 44 : TIM HALVORSEN, JULY 1978
0000 45 :
0000 46 : MODIFIED BY
0000 47 :
0000 48 :
0000 49 : V001 MTR0001 Mike Rhodes 22-Jun-1981
0000 50 : A. Change all CMPW's referencing an MSG$ symbol to CML's.
0000 51 : B. Change default addressing mode to longword.
0000 52 : C. Remove references to SSDAMSGDEF macro.
0000 53 :
0000 54 :--
```

DUMP
V04-000

MEMORY DUMP ROUTINES
DECLARATIONS

H 1

16-SEP-1984 01:30:22 VAX/VMS Macro V04-00
5-SEP-1984 03:32:29 [SDA.SRC]DUMP.MAR;1

Page 3
(2)

**F

0000 56
0000 57 :
0000 58 :
0000 59 :
0000 60
0000 61
0000 62
0000 63

.SBTTL DECLARATIONS

SYMBOL DEFINITIONS

\$DMPDEF
\$VADEF
\$PCBDEF
\$PHDEF

: DUMP FILE DEFINITIONS
: VIRTUAL ADDRESS DEFINITIONS
: PROCESS CONTROL BLOCK
: PROCESS HEADER

0000	65	.SBTTL	STORAGE DEFINITIONS
0000	66		
0000	67		
0000	68		
0000	69		
0000	70		
0000	71		
00000000	72	.PSECT	DUMP,EXE,NOWRT,LONG
0000	73		
0000	74	.DEFAULT	DISPLACEMENT,LONG

.....

```

0000 76 .SBTTL SHOW_SYSTEM - DUMP SYSTEM REGION VIRTUAL MEMORY
0000 77 :---
0C00 78 :
0000 79 : SHOW_SYSTEM
0000 80 :
0000 81 : DISPLAY THE SYSTEM REGION VIRTUAL ADDRESS SPACE
0000 82 :
0000 83 : INPUTS:
0000 84 :
0000 85 : NONE
0000 86 :
0000 87 : OUTPUTS:
0000 88 :
0000 89 : NONE
0000 90 :
0000 91 :---
0000 92 :
0000 93 SHOW_SYSTEM::
0004 0000 94 .WORD ^M<R2>
0002 95
0002 96 SUBHD <System region memory>
000F 97 SKIP PAGE
7E 00000000'EF 1F 01 DD 0016 98 PUSHL #1 ; ENABLE PAGING
00000000'EF 7E 01 1F 78 0021 100 EXTV #0,#31,MMG$FRSTRONLY,-(SP) ; LENGTH OF DUMP
000000DB'EF 03 FB 0025 101 ASHL #VASV SYSTEM,#1,-(SP) ; STARTING ADDRESS
000000DB'EF 03 FB 0025 101 CALLS #3,DUMP ; DUMP UP TO READ-ONLY CODE
000000DB'EF 03 FB 002C 102 SKIP PAGE
01 DD 0033 103 PUSHL #1 ; ENABLE PAGING
1C 50 E9 0045 105 GETMEM @MMG$GL_SPTBASE,-(SP) ; GET ADDRESS OF SPT
0C 50 E9 0048 106 BLBC R0,90$ ; SKIP REST IF ERROR
6E 51 C2 0055 107 GETMEM @EXE$GL_INTSTK ; ADDRESS OF INTERRUPT STACK
000000DB'EF 51 DD 0058 108 BLBC R0,90$ ; SKIP REST IF ERROR
000000DB'EF 51 DD 005B 109 SUBL R1,(SP) ; LENGTH OF BALANCE SLOTS+SYSPHD
000000DB'EF 03 FB 005D 110 PUSHL R1 ; ADDRESS OF BALANCE SLOTS
0064 111 90$: CALLS #3,DUMP ; DUMP DYNAMIC SYSTEM REGION
0064 112 STATUS SUCCESS
04 006B 113 RET

```


006C 115 .SBTTL SHOW_PO -- SHOW PO VIRTUAL MEMORY

006C 116 :---

006C 117 :
006C 118 SHOW_PO

006C 119 :
006C 120 DISPLAY PO VIRTUAL ADDRESS SPACE

006C 121 :
006C 122 INPUTS:

006C 123 :
006C 124 NONE

006C 125 :
006C 126 OUTPUTS:

006C 127 :
006C 128 NONE

006C 129 :
006C 130 :---

006C 131 :
006C 132 SHOW_PO::
0004 006C 133 .WORD ^M<R2>

006E 134 :
006E 135 SUBHD <Process region memory>

01 DD 007B 136 SKIP PAGE

7E 00000000'EF 01 DD 0082 137 PUSHL #1 ; ENABLE PAGING

09 78 0C84 138 : MOVL #<80@9>,R2 ; MAXIMUM TO PRINT (80 PAGES)

0084 139 : ASHL #9,POLR,-(SP) ; LENGTH OF PROCESS REGION

008C 140 : CMPL (SP),R2 ; CHECK IF OVER MAXIMUM

008C 141 : BLEQ 10\$; BRANCH IF OK

008C 142 : MOVL R2,(SP) ; SET TO MAXIMUM

008C 143 10\$:

00 DD 008C 144 PUSHL #0 ; STARTING ADDRESS

03 FB 008E 145 CALLS #3,DUMP ; DUMP PROCESS REGION

04 0095 146 RET

```

0096 148 .SBTTL SHOW_P1 -- DISPLAY P1 VIRTUAL MEMORY
0096 149 :---;
0096 150 :
0096 151 SHOW_P1
0096 152 :
0096 153 PRINT P1 VIRTUAL ADDRESS SPACE
0096 154 :
0096 155 INPUTS:
0096 156 :
0096 157 NONE
0096 158 :
0096 159 OUTPUTS:
0096 160 :
0096 161 NONE
0096 162 :
0096 163 :---
0096 164 :
0096 165 .ENABL LSB
0096 166 :
0004 0096 167 SHOW_P1::
0098 168 .WORD ^M<R2>
0098 169 :
0098 170 SUBHD <Control region memory>
00A5 171 SKIP PAGE
51 00000000'EF 01 DD 00AC 172 PUSHL #1 ; ENABLE PAGING
OC C1 00AE 173 ADDL3 #12,CTL$AL_STACK,R1 ; ADDRESS OF USPINI
11 50 E9 00B6 174 GETMEM (R1) ; GET USPINI
00BF 175 BLBC R0,90$ ; SKIP REST IF ERROR
7E 80000000 8F 51 C3 00C2 176 : MOVL #<80@y>,R2 ; MAXIMUM TO PRINT (80 PAGES)
00C2 177 SUBL3 R1,#^X80000000,-(SP) ; LENGTH
00CA 178 : CMPL (SP),R2 ; CHECK IF OVER MAXIMUM
00CA 179 : BLEQ 10$ ; BRANCH IF OK
00CA 180 : MOVL R2,(SP) ; SET TO MAXIMUM
00CA 181 10$:
000000DB'EF 51 DD 00CA 182 PUSHL R1 ; STARTING ADDRESS
03 FB 00CC 183 CALLS #3,DUMP ; DUMP THE CONTROL REGION
00D3 184 90$:
00D3 185 STATUS SUCCESS
04 00DA 186 RET
00DB 187 :
00DB 188 .DSABL LSB

```

30
01
03
01

49
44

2F

```

00DB 190 .SBTTL DUMP - DUMP SPECIFIED MEMORY AREA
00DB 191 :---
00DB 192 :
00DB 193 DUMP
00DB 194 :
00DB 195 THIS ROUTINE IS CALLED TO PRODUCE A READABLE DISPLAY
00DB 196 OF A SPECIFIED AREA OF MEMORY ON A PRINTER PAGE. ZEROS
00DB 197 ARE SUPPRESSED AND A MESSAGE IS GIVEN INDICATING THE
00DB 198 EXTENT OF THE ZERO AREA. THE MEMORY IS SHOWN IN BOTH
00DB 199 HEXIDECIMAL AND ASCII.
00DB 200 :
00DB 201 INPUTS:
00DB 202 :
00DB 203 4(AP) = STARTING ADDRESS OF THE BUFFER
00DB 204 8(AP) = LENGTH OF THE BUFFER (BYTES)
00DB 205 12(AP) = TRUE IF PAGING ENABLED, ELSE FALSE
00DB 206 :
00DB 207 OUTPUTS:
00DB 208 :
00DB 209 THE MEMORY IS DUMPED STARTING WITH THE FIRST MEMORY
00DB 210 LOCATION. NO SPECIAL HEADERS ARE PRINTED.
00DB 211 :
00DB 212 :---
00DB 213 :
00DB 214 .ENABL LSB
00DB 215 :
00DB 216 DUMP::
0074 00DB 217 .WORD ^M<R2,R4,R5,R6>
00DD 218 :
52 56 20 D0 00DD 219 MOVL #32,R6 ; ASSUME 32 BYTES/LINE
51 1F D0 00E0 220 MOVL #^X1F,R1 ; 32 BYTE GRANULATION
00000000'EF DE 00E3 221 MOVAL BUFFER,R2
00000000'EF D5 00EA 222 TSTL OUTPUT_FILE ; OUTPUT LISTING SPECIFIED?
06 12 00F0 223 BNEQ 5$ ; BRANCH IF YES
56 10 D0 00F2 224 MOVL #16,R6 ; ONLY 16 BYTES/LINE FOR TERMINALS
51 0F D0 00F5 225 MOVL #^XF,R1 ; 16 BYTE GRANULATION
00F8 226 5$:
54 7C 00F8 227 CLRQ R4 ; SET CURRENT STATE TO NORMAL
00FA 228 BICL R1,4(AP) ; BACKUP TO LAST 32 BYTE BOUNDARY
08 AC 51 C0 00FA 229 ADDL R1,8(AP) ; ROUND LIMIT TO NEXT 32 BYTES
08 AC 51 CA 00FE 230 BICL R1,8(AP)
01 14 0102 231 BGTR 10$ ; BRANCH IF NON-ZERO
04 0104 232 RET
0105 233 10$:
07 50 E8 0113 234 TRYMEM @4(AP),(R2),R6 ; GET NEXT N BYTES FROM DUMP
50 01 D0 0116 235 BLBS R0,20$ ; BRANCH IF MEMORY OK
69 10 0119 236 MOVL #1,R0
17 11 011B 237 BSBB DUMP_STATE ; SET STATE = 1 (BYPASS INVALID MEMORY)
011D 238 BRB 60$ ; AND SKIP THIS LOOP
62 56 00 3B 011D 239 20$:
07 12 0121 240 SKPC #0,R6,(R2) ; CHECK IF NON-ZERO BUFFER
50 02 D0 0123 241 BNEQ 30$ ; BRANCH IF NON-ZERO
5C 10 0126 242 MOVL #2,R0
0A 11 0128 243 BSBB DUMP_STATE ; SET STATE = 2 (BYPASS ZEROS)
012A 244 BRB 60$ ; AND SKIP THIS LOOP
50 D4 012A 245 30$:
012A 246 CLRL R0

```

51	04	AC	56	10	012C	247	BSBB	DUMP_STATE	:	SET STATE TO NORMAL
			D0	012E	248		MOVL	4(AP),R1	:	STARTING VIRTUAL ADDRESS
			12	10	0132	249	BSBB	DUMP_LINE	:	DUMP A SINGLE LINE
					0134	250				
04	AC		56	C0	0134	251	ADDL2	R6,4(AP)	:	INCREMENT CURRENT ADDRESS
08	AC		56	C2	0138	252	SUBL2	R6,8(AP)	:	DECREMENT LENGTH
			C7	14	013C	253	BGTR	10\$:	CONTINUE UNTIL DONE
			50	D4	013E	254	CLRL	R0	:	
			42	10	0140	255	BSBB	DUMP_STATE	:	TERMINATE CURRENT STATE (IF ANY)
50		01		D0	0142	256	MOVL	#1,R0	:	SUCCESS
				04	0145	257	RET		:	
					0146	258			:	
					0146	259	.DSABL	LSB	:	

```

0146 262 :---
0146 263 :
0146 264 :      SUBROUTINE TO PRINT A SINGLE LINE OF A MEMORY DUMP
0146 265 :
0146 266 :      R1 = STARTING VIRTUAL ADDRESS OF MEMORY
0146 267 :      R2 = ADDRESS OF LOCAL COPY OF MEMORY
0146 268 :      R6 = LENGTH TO DUMP (EITHER 16 OR 32 BYTES)
0146 269 :      12(AP) = TRUE IF PAGING ENABLED, ELSE FALSE
0146 270 :---
0146 271 :      .ENABL  LSB
0146 272 :
0146 273 DUMP_LINE:
51      DD 0146 274 PUSHL  R1          ; DUMP ADDRESS
52      DD 0148 275 PUSHL  R2          ; ADDRESS OF MEMORY
56      DD 014A 276 PUSHL  R6          ; LENGTH OF STRING
51 56 50 FE 8F 78 014C 277 MOVL   R2,R0
80      DD 0154 278 ASHL   #-2,R6,R1    ; REPEAT COUNT (# LONGWORDS)
51      FE 8F 78 014F 278 40$: PUSHL (R0)+
OF 56 05 05 E0 0154 279 40$: PUSHL (R0)+
51      FE 8F 78 0156 280 SOBGTR R1,40$    ; PUSH LONGWORDS
OF 56 05 05 E0 0159 281 BBS    #5,R6,50$  ; BRANCH IF 32 BYTES/LINE
05      OD 11 015D 282 PRINT  7,<!XL !XL !XL !XL !AF !XL>
05      OD 11 016A 283 BRB    60$
05      OD 11 016C 284 50$: PRINT 11,<!XL !XL !XL !XL !XL !XL !XL !XL !AF !XL>
06 OC AC E8 0179 285 60$: BLBS  12(AP),90$  ; BRANCH IF PAGING ENABLED
00000000'EF D4 017D 286 60$: CLRL  LINE_COUNT ; ELSE INHIBIT PAGE EJECTS
05      OD 05 0183 287 90$: RSB
0184 288
0184 289 .DSABL  LSB

```

: R

.....

.....

.....

.....

.....
S
R
C

```

0184 292 :---
0184 293 :
0184 294 :   SUBROUTINE TO SET STATE OF DUMPING PROCESS
0184 295 :
0184 296 :   R0 = STATE NUMBER TO ENTER (IF NOT ALREADY THERE)
0184 297 :       0 = NORMAL MEMORY
0184 298 :       1 = BYPASS INVALID MEMORY
0184 299 :       2 = ZERO MEMORY
0184 300 :   R4 = CURRENT STATE
0184 301 :   R5 = NUMBER OF BYTES BYPASSED IN CURRENT STATE
0184 302 :   R6 = BYTES DISPLAYED PER LINE
0184 303 :---
0184 304 :
0184 305 :   .ENABL  LSB
0184 306 :
0184 307 :   DUMP_STATE:
54  50  D1 0184 308 :   CMPL  R0,R4           : CHECK IF ALREADY IN STATE
   04  12 0187 309 :   BNEQ  10$           : BRANCH IF NOT
55  56  C0 0189 310 :   ADDL2 R6,R5         : INCREMENT BYPASS COUNTER
   05  05 018C 311 :   RSB
   018D 312 10$:
   018D 313 :   PUSHL R0           : SAVE NEW STATE
01  54  D1 018F 314 :   CMPL  R4,#1         : CHECK IF BYPASSING INVALID MEMORY
   2B  12 0192 315 :   BNEQ  20$           : BRANCH IF NOT
   0194 316 :   SKIP  1
7E  04  AC 01  C3 019D 317 :   SUBL3 #1,4(AP),-(SP) : ENDING ADDRESS
7E  04  AC 55  C3 01A2 318 :   SUBL3 R5,4(AP),-(SP) : STARTING ADDRESS
   01A7 319 :   PRINT 2,<Virtual locations !XL through !XL are not in physical memory>
   01B4 320 :   SKIP  1
   5A  11 01BD 321 :   BRB  80$
   01BF 322 20$:
02  54  D1 01BF 323 :   CMPL  R4,#2         : CHECK IF BYPASSING ZEROS
   55  12 01C2 324 :   BNEQ  80$           : BRANCH IF NOT
30  55  D1 01C4 325 :   CMPL  R5,#48        : ONLY SHOW IF MORE THAN 48 BYTES
   2B  15 01C7 326 :   BLEQ  25$          : BRANCH IF LESS (NOT WORTH MESSAGE)
   01C9 327 :   SKIP  1
7E  04  AC 01  C3 01D2 328 :   SUBL3 #1,4(AP),-(SP) : ENDING ADDRESS
7E  04  AC 55  C3 01D7 329 :   SUBL3 R5,4(AP),-(SP) : STARTING ADDRESS
   01DC 330 :   PRINT 2,<Zeros suppressed from !XL through !XL>
   01E9 331 :   SKIP  1
   25  11 01F2 332 :   BRB  80$
   01F4 333 25$:
   52  DD 01F4 334 :   PUSHL R2           : SAVE ORIGINAL BUFFER ADDRESS
5E  56  C2 01F6 335 :   SUBL  R6,SP         : ALLOCATE BUFFER
52  5E  D0 01F9 336 :   MOVL  SP,R2         : SET ADDRESS OF BUFFER
   3C  BB 01FC 337 :   PUSHR #*M<R2,R3,R4,R5> : SAVE REGISTERS
62  56  00 6E  00 2C 01FE 338 :   MOVCS #0,(SP),#0,R6,(R2) : ZERO THE BUFFER
   3C  BA 0204 339 :   POPR  #*M<R2,R3,R4,R5> : RESTORE REGISTERS
   55  C3 0206 340 28$:
   FF 38 30 020B 341 :   SUBL3 R5,4(AP),R1   : STARTING VIRTUAL ADDRESS
   55  56  C2 020E 342 :   BSBW DUMP LINE     : DUMP A SINGLE LINE OF ZEROS
   F3  14 0211 343 :   SUBL  R6,R5        : DECREMENT AMOUNT LEFT TO DO
   5E  56  C0 0213 344 :   BGTR  28$         : CONTINUE UNTIL DONE
   52  8ED0 0216 345 :   ADDL  R6,SP        : DEALLOCATE BUFFER
   0219 346 80$:
   54  8ED0 0219 347 :   POPL  R2           : RESTORE BUFFER ADDRESS
   55  56  D0 021C 348 :   POPL  R4           : SET NEW STATE
   :   MOVL  #6,R5       : INITIALIZE BYPASS COUNTER

```

DUMP
V04-000

MEMORY DUMP ROUTINES D 2
DUMP - DUMP SPECIFIED MEMORY AREA
05 021F 349 RSB
0220 350
0220 351 .DSABL LSB

16-SEP-1984 01:30:22 VAX/VMS Macro V04-00
5-SEP-1984 03:32:29 [SDA.SRC]DUMP.MAR;1

Page 12
(9)

HAN
Tab

DUMP
V04-000

MEMORY DUMP ROUTINES
DUMP - DUMP SPECIFIED MEMORY AREA

E 2

16-SEP-1984 01:30:22 VAX/VMS Macro V04-00
5-SEP-1984 03:32:29 [SDA.SRC]DUMP.MAR;1

Page 13
(11)

HAN
V04

0220 353
0220 354 .END

DUMP
Symbol table

MEMORY DUMP ROUTINES

F 2

16-SEP-1984 01:30:22 VAX/VMS Macro V04-00
5-SEP-1984 03:32:29 [SDA.SRC]DUMP.MAR;1

Page 14
(11)

HAN
V04

```

ARGS          = 00000003
BUFFER        ***** X 02
CTLSAL_STACK ***** X 02
DUMP          000000DB RG 02
DUMP_LINE     00000146 R 02
DUMP_STATE    00000184 R 02
EXESGL_INTSTK ***** X 02
GETMEM        ***** X 02
LINE_COUNT    ***** X 02
MMGSFRSTRONLY ***** X 02
MMGSGL_SPTBASE ***** X 02
MSG$ SUCCESS  ***** X 02
NEW PAGE      ***** X 02
OUTPUT_FILE   ***** X 02
POLR          ***** X 02
PRINT         ***** X 02
SET HEADING    ***** X 02
SHOW_PO       0000006C RG 02
SHOW_P1       00000096 RG 02
SHOW_SYSTEM   00000000 RG 02
SKIP_LINES    ***** X 02
TRYMEM        ***** X 02
VASV_SYSTEM   = 0000001F
  
```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
DUMP	00000220 (544.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG
LITERALS	0000011F (287.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	38	00:00:00.04	00:00:01.89
Command processing	134	00:00:00.43	00:00:04.78
Pass 1	202	00:00:02.76	00:00:13.99
Symbol table sort	0	00:00:00.31	00:00:01.22
Pass 2	76	00:00:00.73	00:00:03.23
Symbol table output	4	00:00:00.02	00:00:00.02
Psect synopsis output	1	00:00:00.02	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	457	00:00:04.33	00:00:25.17

The working set limit was 1350 pages.
 24229 bytes (48 pages) of virtual memory were used to buffer the intermediate code.
 There were 20 pages of symbol table space allocated to hold 322 non-local and 25 local symbols.
 354 source lines were read in Pass 1, producing 17 object records in Pass 2.
 18 pages of virtual memory were used to define 17 macros.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[SDA.OBJ]SDALIB.MLB;1	7
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	3
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	14

437 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:DUMP/OBJ=OBJ\$:DUMP MSRCS\$:DUMP/UPDATE=(ENHS\$:DUMP)+EXECMLS\$/LIB+LIB\$:SDALIB/LIB

A grid of 14 columns and 10 rows of small, low-contrast images. Each image is a thumbnail of a document page, likely a manual or technical specification. The pages contain various text, diagrams, and tables. Several pages are clearly labeled with the following text:

- HANDLER LIS
- MAPPING LIS
- DUMP LIS
- MAIN LIS
- EXAMPS LIS
- MMG LIS
- INDEX LIS
- LOCK LIS
- PARSE LIS

The overall appearance is that of a microfiche card, where the grid of images represents individual frames of data.