



```
TTTTTTTTT1  IIIIII  MM      MM  EEEEEEEEE  RRRRRRRR
TTTTTTTTTT  IIIIII  MM      MM  EEEEEEEEE  RRRRRRRR
TT          II      MMMM   MMMM  EE          RR          RR
TT          II      MMMM   MMMM  EE          RR          RR
TT          II      MM     MM    EE          RR          RR
TT          II      MM     MM    EE          RRRRRRRR
TT          II      MM     MM    EEEEEEEEE  RRRRRRRR
TT          II      MM     MM    EE          RR  RR
TT          II      MM     MM    EE          RR  RR
TT          II      MM     MM    EE          RR  RR
TT          IIIIII  MM     MM  EEEEEEEEE  RR          RR
TT          IIIIII  MM     MM  EEEEEEEEE  RR          RR
```

```
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          IIIIII  SSSSSSSS
LLLLLLLLLLL IIIIII  SSSSSSSS
```

(2)	47	DECLARATIONS	
(3)	87	MACSTIMER_ON	BEGIN TIMING A BLOCK OF CODE
(4)	119	MACSTIMER_OFF	FINISH TIMING A BLOCK OF CODE

```

0000 1      .TITLE  MACSTIMER      TIMER ROUTINES
0000 2      .IDENT  'V04-000'
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 :++
0000 30 : FACILITY:      VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35 : modules for input to the VAX-11 LINKER.
0000 36 :
0000 37 : ENVIRONMENT:  USER MODE
0000 38 :
0000 39 : AUTHOR: Benn Schreiber, CREATION DATE: 20-AUG-78
0000 40 :
0000 41 : MODIFIED BY:
0000 42 :
0000 43 :      : VERSION
0000 44 : 01  -
0000 45 :--

```

```
0000 47 .SBTTL DECLARATIONS
0000 48 :
0000 49 : INCLUDE FILES:
0000 50 :
0000 51 :
0000 52 :
0000 53 : MACROS:
0000 54 :
0000 55 :
0000 56 $JPIDEF ;DEFINE JOB/PROCESS INFO ITEM CODES
0000 57
0000 58
0000 59 :
0000 60 : LOCAL STORAGE
0000 61 :
0000 62
00000000 63 .PSECT MACSTIMER_DATA,NOEXE,LONG
0000 64
0000 65 GETJPI_ARGBLK: ;ARGUMENT BLOCK FOR GETJPI
0000 66 $GETJPI ITMLST=GETJPI_LIST
0020 67
0020 68 GETJPI_LIST:
0407 0008 0020 69 .WORD 8,JPIS$ CPU_TIM ;RESULT IS 8 BYTES LONG
0000003C' 0024 70 .LONG CPU_TIME ;WHERE TO RETURN CPU TIME
00000000 0028 71 .LONG 0 ;DON'T CARE HOW LONG RESULT IS
040A 0004 002C 72 .WORD 4,JPIS$ PAGEFLTS ;4 BYTES, GET # OF PAGE FAULTS
0000004C' 0030 73 .LONG PAGE_FAULTS ;WHERE TO STORE RESULT
00000000 0034 74 .LONG 0 ;DON'T CARE HOW LONG RESULT IS
00000000 0038 75 .LONG 0 ;END OF LIST
003C 76
003C 77 .DEBUG CPU_TIME,TIME,PAGE_FAULTS
003C 78
00000044 003C 79 CPU_TIME:.BLKQ 1 ;CPU TIME RETURNED FROM GETJPI
0044 80 TIME:
0000004C 0044 81 .BLKQ 1
82 PAGE_FAULTS:
00000050 004C 83 .BLKL 1 ;# PAGE FAULTS
0050 84
00000000 85 .PSECT MAC$RO_CODE_P15,NOWRT,GBL,LONG
```

```

0000 87      .SBTTL MACSTIMER_ON  BEGIN TIMING A BLOCK OF CODE
0000 88
0000 89      :++
0000 90      : FUNCTIONAL DESCRIPTION:
0000 91      :
0000 92      : THIS ROUTINE SETS THE STARTING TIME FOR A BLOCK OF CODE.
0000 93      :
0000 94      : CALLING SEQUENCE:
0000 95      :
0000 96      : PUSHAB DUAL_QUAD_WORDS
0000 97      : CALLS #1,MACSTIMER_ON
0000 98      :
0000 99      : INPUTS:
0000 100     :
0000 101     : DUAL_QUAD_WORDS IS THE ADDRESS OF TWO QUADWORDS. THE FIRST
0000 102     : WILL RECEIVE THE CPU TIME FOR THE BLOCK OF
0000 103     : CODE (DELTA FORMAT) AND THE SECOND WILL RECEIVE
0000 104     : THE ELAPSED TIME FOR THE BLOCK OF CODE.
0000 105     :
0000 106     :
0000 107     :--
0000 108
0000 109     MACSTIMER_ON::

```

```

0000 0000 110     .WORD 0
0000 0002 111     $GETJPI_G GETJPI_ARGBLK      ;GET ACCUMULATED CPU TIME
50 04 AC DO 000D 112     MOVL 4(AP),R0      ;GET ADDRESS OF DUAL_QUAD_WORDS
80 003C'CF 7D 0011 113     MOVQ W^CPU_TIME,(R0)+  ;SAVE STARTING CPU TIME
08 A0 004C'CF 9F 0016 114     PUSHAB (R0)      ;STACK ADDRESS OF STARTING ELAPSED TIME
00000000'GF 01 FB 001E 115     MOVL W^PAGE_FAULTS,8(R0) ;STORE STARTING # PAGE FAULTS
04 0025 116     CALLS #1,G^SYSS$GETTIM ;GET STARTING WALL CLOCK TIME
04 0025 117     RET

```

```

0026 119      .SBTTL MACSTIMER_OFF FINISH TIMING A BLOCK OF CODE
0026 120
0026 121      :++
0026 122      : FUNCTIONAL DESCRIPTION:
0026 123      :
0026 124      : THIS ROUTINE COMPUTES THE CPU TIME AND ELAPSED TIME REQUIRED
0026 125      : FOR A BLOCK OF CODE.
0026 126      :
0026 127      : CALLING SEQUENCE:
0026 128      :
0026 129      : PUSHAB DUAL_QUAD_WORD
0026 130      : CALLS #1,MACSTIMER_OFF
0026 131      :
0026 132      : INPUTS:
0026 133      :
0026 134      : DUAL_QUAD_WORD IS THE ADDRESS OF THE SAME DUAL QUAD WORD PASSED
0026 135      : TO MACSTIMER_ON TO START TIMING THE BLOCK.
0026 136      : THE ACCUMULATED CPU TIME IS RETURNED IN THE FIRST
0026 137      : QUAD WORD (DELTA FORMAT) AND THE ELAPSED TIME
0026 138      : IS RETURNED IN THE SECOND QUAD WORD.
0026 139      :
0026 140      :--
0026 141
0026 142      .ALIGN LONG
0028 143      MACSTIMER_OFF::
0600 0028 144      .WORD ^M<R9,R10> ;REGISTER SAVE MASK
002A 145      $GETJPI_G GETJPI ARGBLK ;GET ACCUMULATED CPU TIME
5A 0044'CF 9E 0035 146      MOVAB W^TIME,RTO ;POINT TO QUADWORD
00000000'GF 01 9F 003A 147      PUSHAB (R10) ;STACK QUADWORD ADDRESS
50 04 AC D0 003C 148      CALLS #1,G^SYSS$GETTIM ;GET CURRENT WALL CLOCK TIME
59 003C'CF 9E 0043 149      MOVL 4(AP),R0 ;POINT TO FIRST QUAD WORD
80 89 60 C3 0047 150      MOVAB W^CPU_TIME,R9 ;POINT TO CPU-TIME QUAD WORD
80 69 60 D9 0050 151      SUBL3 (R0),(R9)+,(R0)+ ;COMPUTE ACCUMULATED CPU TIME
80 69 60 D9 0053 152      SBWC (R0),(R9) ;
80 69 60 D0 0056 153      MOVL (R9),(R0)+ ;STORE IN THE QUADWORD
80 8A C2 0059 154      SUBL2 (R10)+,(R0)+ ;FIGURE ELAPSED REAL TIME
60 004C'CF 60 C3 005C 155      SBWC (R10),(R0)+ ;
04 0062 156      SUBL3 (R0),W^PAGE_FAULTS,(R0) ;COMPUTE # OF PAGE FAULTS THIS PHASE
0063 157      RET
0063 158
0063 159      .END

```

```

$$ARGS = 00000007
$$T1 = 00000020
CPU TIME = 0000003C R D 02
GETJPI$ASTADR = 00000018
GETJPI$ASTPRM = 0000001C
GETJPI$EFN = 00000004
GETJPI$IOSB = 00000014
GETJPI$ITMLST = 00000010
GETJPI$NARGS = 00000007
GETJPI$PIDADR = 00000008
GETJPI$PRCNAM = 0000000C
GETJPI$ARGBLK = 00000000 R 02
GETJPI$LIST = 00000020 R 02
JPI$CPUTIM = 00000407
JPI$PAGEFLTS = 0000040A
MAC$TIMER_OFF = 00000028 RG 03
MAC$TIMER_ON = 00000000 RG 03
PAGE FAULTS = 0000004C R D 02
SY$$GETJPI ***** GX 03
SY$$GETTIM ***** X 03
TIME = 00000044 R D 02
    
```

-----  
! 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
MAC\$TIMER_DATA	00000050 ( 80.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
MAC\$RO_CODE_P15	00000063 ( 99.)	03 ( 3.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.03	00:00:01.58
Command processing	127	00:00:00.41	00:00:03.70
Pass 1	139	00:00:01.20	00:00:07.55
Symbol table sort	0	00:00:00.08	00:00:00.08
Pass 2	44	00:00:00.32	00:00:02.07
Symbol table output	4	00:00:00.02	00:00:00.02
Psect synopsis output	1	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	353	00:00:02.07	00:00:15.01

The working set limit was 1050 pages.  
8305 bytes (17 pages) of virtual memory were used to buffer the intermediate code.  
There were 10 pages of symbol table space allocated to hold 128 non-local and 0 local symbols.  
159 source lines were read in Pass 1, producing 16 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:[MACRO.OBJ]MACRO.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	8

199 GETS were required to define 8 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:TIMER/OBJ=OBJ\$:TIMER MSRC\$:TIMER/UPDATE=(ENH\$:TIMER)+LIB\$:MACRO/LIB

