


```

MM      MM      TTTTTTTTTT      AAAAAA      DDDDDDDD      EEEEEEEEEE      FFFFFFFFFF
MM      MM      TTTTTTTTTT      AAAAAA      DDDDDDDD      EEEEEEEEEE      FFFFFFFFFF
MMM     MMM     TT              AA      AA      DD      DD      EE              FF
MMM     MMM     TT              AA      AA      DD      DD      EE              FF
MM      MM      TT              AA      AA      DD      DD      EE              FF
MM      MM      TT              AA      AA      DD      DD      EE              FF
MM      MM      TT              AA      AA      DD      DD      EEEEEEEEE      FFFFFFFF
MM      MM      TT              AA      AA      DD      DD      EEEEEEEEE      FFFFFFFF
MM      MM      TT              AAAAAAAAAA      DD      DD      EE              FF
MM      MM      TT              AAAAAAAAAA      DD      DD      EE              FF
MM      MM      TT              AA      AA      DD      DD      EE              FF
MM      MM      TT              AA      AA      DD      DD      EE              FF
MM      MM      TT              AA      AA      DDDDDDDD      EEEEEEEEEE      FFFFFFFF
MM      MM      TT              AA      AA      DDDDDDDD      EEEEEEEEEE      FFFFFFFF

```

```

88888888      333333      222222
88888888      333333      222222
88      88      33      33      22      22
88      88      33      33      22      22
88      88      33      33      22      22
88      88      33      33      22      22
88888888      33      22
88888888      33      22
88      88      33      22
88      88      33      22
88      88      33      22
88      88      33      22
88888888      333333      2222222222
88888888      333333      2222222222

```



Definition file for MTAACP compilation

Version: 'V04-000'

```

*****
*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****

```

♦♦

FACILITY: MAGNETIC TAPE ACP

ABSTRACT:

These are the data structure definitions and random macros used to compile the MTAACP.

ENVIRONMENT:

Starlet Operating System, including privileged system calls and internal system subroutines.

--

Author: D. H. Gillespie, Creation date: 18-may-77 16:25

MODIFIED BY:

- V03-006 MMD0178 Meg Dumont, 26-May-1983 15:14
Fix to support new input to IOCSVT_DEVNAM
- V03-005 MMD0173 Meg Dumont, 9-May-1983 15:15
Fix to make USER_STATUS consistently defined within module
- V03-004 MMD0145 Meg Dumont, 25-Apr-1983 18:14

Add HDR4 label, add some literals for scratch_offset
and file_spec_max

V03-003 MMD0119 Meg Dumont, 29-Mar-1983 0:44
Added misc def's common inside the MTAACP

V03-002 MMD0002 Meg Dumont, 5-Jan-1983 13:50
Add another field to be define for V3.0 systems

V03-001 MMD0001 Meg Dumont, 11-Nov-1982 10:44
Add VCB def for enable user EOT handling. Needed to work on V3.X

V02-008 DMW00075 David Michael Walp 8-Feb-1982
Added VVPS(T;S)_ACCOUNT and changed _PROC_NAME to _USERNAME

V02-007 DMW00058 David Michael Walp 7-Dec-1981
Removed MAX_FILESTR_LEN

V02-006 DMW00047 David Michael Walp 30-Jul-1981
Removed MAX_ATTR_CODE for global ATRSC_MAX_CODE, added
assume MACRO

V02-005 DMW00028 David Michael Walp 30-Jul-1981
Remove MOUSV_NORDVOL1 Added MOUSV_CHKIFSPC

V02-004 DMW00027 David Michael Walp 20-Jul-1981
Added True and False

V02-003 DMW00020 David Michael Walp 26-May-1981
Added Work Area Sz which was a GLOBAL LITERAL from OPCOM.
Inceased MSGSIZE from 20 to 124, new OPCOM message format.

V02-002 MCN0018 Maria del C. Nasr 24-Jun-1980
Fix syntax error in the linkage definitions.

V02-001 REFORMAT Maria del C. Nasr 17-Jun-1980

A0103 MCN0003 Maria del C. Nasr 01-Oct-1979 14:45
Add HDR3 definition

A0102 MCN0002 Maria del C. Nasr 26-Sep-79 16:05
Change BUG_CHECK macro to use builtin BUGW.

A0102 SPR20439 D. H. Gillespie, 21-Nov-1978 13:21
add number of tape marks spaced

A0101 DGH0003 D. H. Gillespie 17-NOV-1978 10:00
change ERROR() [CODE] to ERROR [CODE] to be compatible with
new BLISS

```
! define common external registers
```

```
MACRO COMMON_REG = CURRENT_VCB = 11 : REF BBLOCK%;
MACRO GLOBAL_REG = CURRENT_VCB = 11%;
```

```
! define commonly used linkage
```

```
LINKAGE
```

```
COMMON_CALL      = CALL : GLOBAL(GLOBAL_REG),
NOPRES           = CALL : GLOBAL(GLOBAL_REG),
                  NOPRESERVE(0,1,2,3,4,5,6,7,8,9,10),
JSB              = JSB  : GLOBAL(GLOBAL_REG),
L$CLOSE_FILE     = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(2,3,4,5,6,7,8,9),
L$GET_REQ        = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(5,6,7,8,9,10),
L$GET_START_HDR = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(2,3,4,5,6,7,8,9),
L$GTNEXT_VOL_RE = JSB  : GLOBAL(GLOBAL_REG),
L$GTNEXT_VOL_WR = JSB  : GLOBAL(GLOBAL_REG),
                  NOPRESERVE(2,3,4,5,6,7,8,9,10),
L$ISSUE_IO       = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(2,3,4,5,6,7,8,9,10),
L$NEXT_VOL_READ = JSB  : GLOBAL(GLOBAL_REG),
                  NOPRESERVE(2,3,4,5,6,7,8,9,10),
L$NEXT_VOL_WRIT = JSB  : GLOBAL(GLOBAL_REG),
                  NOPRESERVE(2,3,4,5,6,7,8,9,10),
L$PRINT_NOT_LAB = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(5,6,7,8,9,10),
L$PRINT_OPR_MSG = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(2,3,4,5,6,7,8,9,10),
L$REPOSITION     = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(2,3,4,5,6,7,8,9,10),
L$WRAP_AROUND    = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(2,3,4,5,6,7,8,9,10),
L$WRITE_HEADER   = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(7,8,9,10),
L$WRITE_TM       = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(2,3,4,5,6,7,8,9,10),
L$WRITE_TRAILER = JSB  : GLOBAL(GLOBAL_REG),
                  NOTUSED(2,3,4,5,6,7,8,9,10),
L$IOC_CVT_DEVNAM = JSB (REGISTER=0,REGISTER=1,REGISTER=4,REGISTER=5; REGISTER=1) :
                  PRESERVE (2,3,4,5,6)
                  NOTUSED (7,8,9,10,11);
```

```
! define macro to extract size
```

```
MACRO $BYTESIZE(OFFSET,POSITION,WIDTH,SIGN) = WIDTH / 8 %;
```

```
! declare psect usage to minimize page breakage.
```

```
PSECT
```

```
OWN      = $LOCKEDD1$,
GLOBAL   = $LOCKEDD1$,
PLIT     = $CODE$ (EXECUTE);
```

```
: Declare VAX built in functions.
```

```
:  
BUILTIN
```

```
CHMU,      : change mode to user ( ERR_EXIT )  
INSQUE,    : insert into queue  
MOVTUC,    : translate strings and check for invalid characters  
MTPR,      : move to privilege register ( SET_IPL )  
REMQUE,    : remove from queue  
ROT;       : rotate longword
```

```
: Structure declarations used for system defined structures to  
: save typing.
```

```
:  
STRUCTURE
```

```
BBLOCK [O, P, S, E; N] =  
  [N]  
  (BBLOCK+O)<P,S,E>.
```

```
BBLOCKVECTOR [I, O, P, S, E; N, BS] =  
  [N*BS]  
  (BBLOCKVECTOR+(O+I*BS))<P,S,E>;
```

```
: assorted macros used in fcp code
```

```
: set processor IPL
```

```
MACRO SET_IPL (LEVEL) = MTPR (%REF (LEVEL), PRS_IPL)%;
```

```
: Declare code that must be locked into the working set.
```

```
MACRO LOCK_CODE  
      PSECT      =  
      CODE      = $LOCKEDC1$,  
      PLIT      = $LOCKEDC1$,  
      OWN       = $LOCKEDD1$,  
      GLOBAL    = $LOCKEDD1$;  
%;
```

```
: ***** NOTE: The following two macros violate the BLISS language definition  
: ***** in that they make use of the value of SP while building the arg list.  
: ***** It is the opinion of the bliss maintainers that this usage is safe  
: ***** from planned future optimizations.
```

```
: Macro to call the change mode to kernel system service.  
: Macro call format is 'KERNEL_CALL (ROUTINE, ARG1, ARG2, ... )'.
```

```
MACRO  
  KERNEL_CALL (R) =  
    BEGIN  
    EXTERNAL ROUTINE SYSS$CMKRNL : ADDRESSING_MODE (ABSOLUTE);  
    BUILTIN SP;  
    SYSS$CMKRNL(R, .SP, %LENGTH-1  
                %IF %LENGTH GTR 1 %THEN, %REMAINING %FI)  
    END%;
```

```
: Macro to call the change mode to exec system service.  
: Macro call format is 'EXEC_CALL (ROUTINE, ARG1, ARG2, ... )'.
```

```
MACRO  
  EXEC_CALL (R) =  
    BEGIN  
    EXTERNAL ROUTINE SYSS$CMEXEC : ADDRESSING_MODE (ABSOLUTE);  
    BUILTIN SP;  
    SYSS$CMEXEC(R, .SP, %LENGTH-1  
                %IF %LENGTH GTR 1 %THEN, %REMAINING %FI)  
    END%;
```

```
: Macro used to signal fatal errors (internal consistency checks).
```

```
MACRO  
  BUG_CHECK (CODE) =  
    BEGIN  
    BUILTIN BUGW;  
    EXTERNAL LITERAL %NAME('BUG$', CODE);
```

```
        BUGW(%NAME('BUGS_',CODE) OR 4);
        END
        %;
```

```
: Macro to signal an error status and continue.
```

```
MACRO
    ERROR [CODE] =
        BEGIN
            EXTERNAL USER_STATUS : VECTOR [2];
            BEGIN
                MAP USER_STATUS : WORD;
                USER_STATUS = CODE;
            END;
        END
        %;
```

```
: Macro to signal an error status and exit.
: Implemented as a call into a change mode to user instruction followed
: by a RET.
```

```
MACRO
    ERR_EXIT (CODE) =
        (CHMU(%REF (%IF %NULL (CODE) %THEN 0 %ELSE CODE %FI));)%;
```

```
: Macro to generate a string with a descriptor.
```

```
MACRO
    DESCRIPTOR (STRING) =
        UPLIT (%CHARCOUNT (STRING),
            UPLIT BYTE (STRING))%;
```

```
: Macro to return the number of actual parameters supplied to a routine
: call.
```

```
MACRO
    ACTUALCOUNT =
        BEGIN
            BUILTIN AP;
            (.AP)<0,8>
        END%;
```

```
: check to see that constants have not changed
: e.g. ASSUME (IRCSC_FIXOVHDSZ + 2, IRCSC_VAROVHDSZ);
```

```
MACRO ASSUME (A,B) =
    %IF %BYTEOFFSET(A) NEQ %BYTEOFFSET(B)
    %THEN %WARN('WARNING CONSTANT HAS CHANGED')
    %FI %;
```


! Random other definitions

! MOUNT_VOL flags

```
MACRO MOUSV_REWIND = 0,0,1,0%: ! rewind on mount
MACRO MOUSV_LBLCHECK = 0,1,1,0%: ! check label
MACRO MOUSV_CHKIFSPC = 0,2,1,0%: ! check label if operator specified
MACRO MOUSV_MOUNTERR = 0,3,1,0%: ! there was an error, force physical mount
```

LITERAL

! some world famous Boolean values

```
TRUE = 1,
FALSE = 0,
```

! these are the structure types

```
FVP_TYPE = 1,
VVP_TYPE = 2,
```

! this is the size of a mailbox message from the operator

```
WORK_AREA_SZ = 128,
MSGSIZE = 124, ! 4 = status, 4 = id, operator text,
! plus %OPCOM line (WORK_AREA_SZ - 4)
```

```
IOEFN = 1, ! event flag for I/O
EFN = 1, ! event flag for I/O
TIMEFN = 3, ! event flag for timer wait
```

```
EXEC_MODE = 1, ! exec_mode value
USER_MODE = 3, ! user_mode access
MAX_DEVNAM_LENGTH = 16, ! Set the maximum length that a devname
! can be with VMS
NO OF SUPPORT_ANSI_LABELS = 4, ! Number of supported ANSI labels
ANSI_CBSZ = 80, ! Size of the ANSI standard label
FILE_SPEC_MAX = 79, ! Maximum file specification length
! for VMS long file names (39.39)
```


Terminal 1	Terminal 2	Terminal 3	Terminal 4	Terminal 5	Terminal 6	Terminal 7	Terminal 8	Terminal 9	Terminal 10	Terminal 11	Terminal 12
Terminal 13	Terminal 14	Terminal 15	Terminal 16	Terminal 17	Terminal 18	Terminal 19	Terminal 20	Terminal 21	Terminal 22	Terminal 23	Terminal 24
Terminal 25	Terminal 26	Terminal 27	Terminal 28	Terminal 29	Terminal 30	Terminal 31	Terminal 32	Terminal 33	Terminal 34	Terminal 35	Terminal 36
Terminal 37	Terminal 38	Terminal 39	Terminal 40	Terminal 41	Terminal 42	Terminal 43	Terminal 44	Terminal 45	Terminal 46	Terminal 47	Terminal 48
Terminal 49	Terminal 50	Terminal 51	Terminal 52	Terminal 53	Terminal 54	Terminal 55	Terminal 56	Terminal 57	Terminal 58	Terminal 59	Terminal 60
Terminal 61	Terminal 62	Terminal 63	Terminal 64	Terminal 65	Terminal 66	Terminal 67	Terminal 68	Terminal 69	Terminal 70	Terminal 71	Terminal 72
Terminal 73	Terminal 74	Terminal 75	Terminal 76	Terminal 77	Terminal 78	Terminal 79	Terminal 80	Terminal 81	Terminal 82	Terminal 83	Terminal 84
Terminal 85	Terminal 86	Terminal 87	Terminal 88	Terminal 89	Terminal 90	Terminal 91	Terminal 92	Terminal 93	Terminal 94	Terminal 95	Terminal 96
Terminal 97	Terminal 98	Terminal 99	Terminal 100	Terminal 101	Terminal 102	Terminal 103	Terminal 104	Terminal 105	Terminal 106	Terminal 107	Terminal 108
Terminal 109	Terminal 110	Terminal 111	Terminal 112	Terminal 113	Terminal 114	Terminal 115	Terminal 116	Terminal 117	Terminal 118	Terminal 119	Terminal 120
Terminal 121	Terminal 122	Terminal 123	Terminal 124	Terminal 125	Terminal 126	Terminal 127	Terminal 128	Terminal 129	Terminal 130	Terminal 131	Terminal 132
Terminal 133	Terminal 134	Terminal 135	Terminal 136	Terminal 137	Terminal 138	Terminal 139	Terminal 140	Terminal 141	Terminal 142	Terminal 143	Terminal 144