

```
; FILE: SysMacs.Text
; MACWORKS Copy
```

```
; SysMacs.Text - System macros include file. All system components
; should include this file of standard system macros.
```

```
; Written by: Bud Tribble 28-Oct-81
```

```
; Modification History:
```

```
;
; AJH 30-Oct-81 - changed MOVE.L to LEA in "__CORE" macro
; AJH 02-Nov-81 - deleted SEEK and old memory management macros
; AJH 15-Nov-81 - added ASYNC option
; alo 2-Feb-82 - added some utility core routines
; LAK 05-Sep-82 - added DrvrInstall, DrvrRemove
; LAK 07-Sep-82 - added 5 object manager core routines
; MPH 29-Oct-82 - added System/Current option to mem man. macros.
; LAK 23 Dec 82 - removed object manager calls; added new file system
; traps (_UnMountVol,_MountVol,_FlushVol,_FlushFile,
; _OpenRF) and options (FilNam,FilNum);
; removed file system traps _ReadLabel,_WriteLabel.
; LAK 12 Jan 83 added CmpString, GetTrapAddr, SetTrapAddr macros
; LAK 18 Jan 83 added SetFillLock, RstFillLock, SetFillType macros
; LAK 21 Jan 83 made consistent with Andy's SysMacs
; MPH 2 Feb 83 reorganized macros to add new memory manager calls,
; remove SetHLock, SetHPurge
; LAK 4 Feb 83 added Delay (again) and AddDrive macros. Changed
; Exit to LoadFile.
; LAK 9 Feb 83 added SetGrowZone, CompactMem, PurgeMem
; LAK 10 Feb 83 added RDrvrInstall
; LAK 22 Feb 83 trap 63 is now InitUtil.
; AJH 12 Apr 83 Added REGS to GetTrapAddr
; LAK 03 Jun 83 Removed NOSAVE option (no longer used); added SetAppBase.
; LAK 16 Jun 83 Fixed macros for FreeMem, MaxMem, DisposPtr, SetPtrSize,
; GetPtrSize, DisposHandle, SetHandleSize, GetHandleSize,
; ReAllocHandle, EmptyHandle, SetApplLimit, HeapValidate,
; CompactMem, and PurgeMem to reflect correct options, reg save.
; SC 23 Jun 83 Added IMMED option for control calls.
; LAK 06 Aug 83 Added ResrvMem trap in place of HeapValidate; CompactMem,
; FreeMem, PurgeMem, and MaxMem traps adjusted for ,SYS
; option.
; LAK 12 Aug 83 Added ,CLEAR option for memory manager; added ,CASE and
; ,MARKS for string compares.
; LAK 11 Sep 83 Made CompactMem, FreeMem, PurgeMem, and ResrvMem use SAVE
; instead of REGS.
; LAK 16 Nov 83 Added OffLine trap for file system.
; LAK 30 Nov 83 Added MoreMasters call for memory manager.
```

```
;
; Macro TrapTo is used to call the hardware interface routines.
; Parameter %1 is the magic number associated with the routine begin called.
```

```
.MACRO TRAPTO ; <routine>
MOVEM.L D5/D7,-(SP) ; save register
MOVE.W #%1,D7 ; routine number
TRAP #5 ; trap
MOVEM.L (SP)+,D5/D7 ; restore register
.ENDM
```

```
; Misc. system macros
```

```
.MACRO      _CORE
  .IF      '%1' <> ''
  LEA     %1,A0
  .ENDC

  .IF      '%3' = ''
  .WORD   $A000+%2
  .ENDC

  .IF      '%3' = 'REGS'
  .WORD   $A100+%2
  .ENDC

  .IF      '%3' = 'ASYNC'
  .WORD   $A400+%2
  .ENDC

  .IF      '%3' = 'SYS'
  .WORD   $A400+%2
  .ENDC

  .IF      '%3' = 'IMMED'
  .WORD   $A200+%2
  .ENDC

  .ENDM

.MACRO      _MCore
  .IF      '%1' <> ''
  LEA     %1,A0
  .ENDC

  .IF      '%3' = ''
  .WORD   $A000+%2
  .ENDC

  .IF      '%3' = 'SAVE'
  .IF      '%4' = 'SYS'
  .WORD   $A400+%2
  .ELSE
  .WORD   $A000+%2
  .ENDC
  .ENDC

  .IF      '%3' = 'REGS'
  .IF      '%4' = 'SYS'
  .IF      '%5' = 'CLEAR'
  .WORD   $A700+%2
  .ELSE
  .WORD   $A500+%2
  .ENDC
  .ELSE
  .IF      '%4' = 'CLEAR'
  .WORD   $A300+%2
  .ELSE
  .WORD   $A100+%2
  .ENDC
  .ENDC

  .ENDC
```

```
.ENDM
```

```
; International string comparisons have two boolean parameters:
; bit 9 = 0 if and only if diacritical marks are significant
; bit 10 = 0 if and only if case is insignificant
;
```

```
.MACRO   _StrCore
  .IF    '%1' <> ''
    LEA  %1,A0
  .ENDC

  .IF    '%3' = ''
    .WORD $A000+%2
  .ENDC

  .IF    '%3' = 'MARKS'
    .IF    '%4' = 'CASE'
      .WORD $A600+%2
    .ELSE
      .WORD $A200+%2
    .ENDC
  .ENDC

  .IF    '%3' = 'CASE'
    .WORD $A400+%2
  .ENDC

.ENDM
```

```
; Core routine system macros
;
```

```
; First we have the I/O core routine macros. These are also used by
; the file system.
```

```
.MACRO   _Open
  _CORE %1,0,%2
.ENDM
```

```
.MACRO   _Close
  _CORE %1,1,%2
.ENDM
```

```
.MACRO   _Read
  _CORE %1,2,%2
.ENDM
```

```
.MACRO   _Write
  _CORE %1,3,%2
.ENDM
```

```
.MACRO   _Control
  _CORE %1,4,%2
.ENDM
```

```
.MACRO  _Status  
_CORE  %1,5,%2  
.ENDM
```

```
.MACRO  _KillIO  
_CORE  %1,6,%2  
.ENDM
```

; other file system calls go here

```
.MACRO  _GetVolInfo  
_CORE  %1,7,%2  
.ENDM
```

```
.MACRO  _Create  
_CORE  %1,8,%2  
.ENDM
```

```
.MACRO  _Delete  
_CORE  %1,9,%2  
.ENDM
```

```
.MACRO  _OpenRF  
_CORE  %1,10,%2  
.ENDM
```

```
.MACRO  _ReName  
_CORE  %1,11,%2  
.ENDM
```

```
.MACRO  _GetFileInfo  
_CORE  %1,12,%2  
.ENDM
```

```
.MACRO  _SetFileInfo  
_CORE  %1,13,%2  
.ENDM
```

```
.MACRO  _UnMountVol  
_CORE  %1,14  
.ENDM
```

```
.MACRO  _MountVol  
_CORE  %1,15  
.ENDM
```

```
.MACRO  _Allocate  
_CORE  %1,16,%2  
.ENDM
```

```
.MACRO  _GetEOF  
_CORE  %1,17,%2  
.ENDM
```

```
.MACRO  _SetEOF  
_CORE  %1,18,%2  
.ENDM
```

```
.MACRO  _FlushVol  
_CORE  %1,19,%2  
.ENDM
```

```
.MACRO   _GetVol
  _CORE  %1,%2,%2
.ENDM
```

```
.MACRO   _SetVol
  _CORE  %1,%2,%2
.ENDM
```

```
.MACRO   _InitQueue
  _CORE  %1,%2
.ENDM
```

```
.MACRO   _Eject
  _CORE  %1,%2,%2
.ENDM
```

```
.MACRO   _GetFPos
  _CORE  %1,%2,%2
.ENDM
```

; Here are the storage manager core routine macros

```
.MACRO   _InitZone
  _CORE  %1,%25
.ENDM
```

```
.MACRO   _GetZone
  _CORE  %1,%26,REGS
.ENDM
```

```
.MACRO   _SetZone
  _CORE  %1,%27
.ENDM
```

```
.MACRO   _FreeMem
  _MCORE %1,%28,SAVE,%2
.ENDM
```

```
.MACRO   _MaxMem
  _MCORE %1,%29,REGS,%2
.ENDM
```

```
.MACRO   _NewPtr
  _MCORE %1,%30,REGS,%2,%3
.ENDM
```

```
.MACRO   _DisposPtr
  _CORE  %1,%31
.ENDM
```

```
.MACRO   _SetPtrSize
  _CORE  %1,%32
.ENDM
```

```
.MACRO   _GetPtrSize
  _CORE  %1,%33
.ENDM
```

```
.MACRO   _NewHandle
  _MCORE %1,%34,REGS,%2,%3
.ENDM
```

```
.MACRO  _DisposHandle
__CORE  %1,35
.ENDM

.MACRO  _SetHandleSize
__CORE  %1,36
.ENDM

.MACRO  _GetHandleSize
__CORE  %1,37
.ENDM

.MACRO  _HandleZone
__CORE  %1,38,REGS
.ENDM

.MACRO  _ReAllocHandle
__CORE  %1,39
.ENDM

.MACRO  _RecoverHandle
__MCORE %1,40,REGS,%2,%3
.ENDM

.MACRO  _HLock
__CORE  %1,41
.ENDM

.MACRO  _HUnLock
__CORE  %1,42
.ENDM

.MACRO  _EmptyHandle
__CORE  %1,43
.ENDM

.MACRO  _InitApplZone
__CORE  %1,44
.ENDM

.MACRO  _SetApplLimit
__CORE  %1,45
.ENDM

.MACRO  _BlockMove
__CORE  %1,46
.ENDM
```

; Here are the event manager macros

```
.MACRO  _PostEvent
__CORE  %1,47
.ENDM

.MACRO  _OSEventAvail
__CORE  %1,48
.ENDM

.MACRO  _GetOSEvent
__CORE  %1,49
.ENDM
```

```
.MACRO _FlushEvents
  _CORE %1,50
.ENDM
```

; Here are the utility core routine macros

```
.MACRO _VInstall
  _CORE %1,51
.ENDM
```

```
.MACRO _VRemove
  _CORE %1,52
.ENDM
```

```
.MACRO _Offline
  _CORE %1,53,%2
.ENDM
```

```
.MACRO _MoreMasters
  _CORE %1,54
.ENDM
```

```
.MACRO _ReadParam
  _CORE %1,55
.ENDM
```

```
.MACRO _WriteParam
  _CORE %1,56
.ENDM
```

```
.MACRO _ReadDateTime
  _CORE %1,57
.ENDM
```

```
.MACRO _SetDateTime
  _CORE %1,58
.ENDM
```

```
.MACRO _Delay
  _CORE %1,59
.ENDM
```

```
.MACRO _CmpString
  _StrCore %1,60
.ENDM
```

```
.MACRO _DrvInstall
  _CORE %1,61
.ENDM
```

```
.MACRO _DrvRemove
  _CORE %1,62
.ENDM
```

```
.MACRO _InitUtil
  _CORE %1,63
.ENDM
```

```
.MACRO _ResrvMem
  _MCORE %1,64,SAVE,%2
.ENDM
```

```
.MACRO _SetFilLock
  _CORE %1,%2
.ENDM
```

```
.MACRO _RstFilLock
  _CORE %1,%2
.ENDM
```

```
.MACRO _SetFilType
  _CORE %1,%2
.ENDM
```

```
.MACRO _SetFPos
  _CORE %1,%2
.ENDM
```

```
.MACRO _FlushFil
  _CORE %1,%2
.ENDM
```

```
.MACRO _GetTrapAddress
  _CORE %1,REGS
.ENDM
```

```
.MACRO _SetTrapAddress
  _CORE %1,%2
.ENDM
```

```
.MACRO _PtrZone
  _CORE %1,REGS
.ENDM
```

```
.MACRO _HPurge
  _CORE %1,%2
.ENDM
```

```
.MACRO _HNoPurge
  _CORE %1,%2
.ENDM
```

```
.MACRO _SetGrowZone
  _CORE %1,%2
.ENDM
```

```
.MACRO _CompactMem
  _MCORE %1,%2,SAVE,%2
.ENDM
```

```
.MACRO _PurgeMem
  _MCORE %1,%2,SAVE,%2
.ENDM
```

```
.MACRO _AddDrive
  _CORE %1,%2
.ENDM
```

```
.MACRO _RDrvInstall
  _CORE %1,%2
.ENDM
```

```
.MACRO _UprString
```



```
_StrCore %1,84  
.ENDM
```

```
.MACRO _SetAppBase  
_CORE %1,87  
.ENDM
```