

=====
***** FinderFileLoader *****
- An Apple IIgs GS/OS Finder Multi-File Loader Init -
for AppleWorks 5.1
by Hugh Hood
version 1.3 (2024)

based on the AppleWorks 3/UltraMacros 3 'FinderLaunch'
by John L. Tegelaar & Ton W. van Santen
The Netherlands (1991)

=====
Ultra 4 Macro Source Code
=====

This file contains the source code for the Ultra 4 macro portion of the FinderFileLoader Init for AppleWorks 5.1.

These macros in their compiled form are loaded by the FinderFileLoader Init, when AppleWorks is started, to the macro table beginning at \$EF00 in auxiliary memory after the Init first determines whether the machine is running under GS/OS on a Apple IIgs.

AppleWorks then executes the macros, which perform the following tasks:

(i) Place in page 2 (\$200+) and call the code necessary to begin accessing the GS/OS Message Center, where the GS/OS Finder has placed the pathnames of the AppleWorks (and optionally text) files selected by the user;

(ii) Place in page 2 (\$200+) and call the code necessary to retrieve the pathnames of the selected filenames from the GS/OS Message Center;

(iii) Load each file in turn to the AppleWorks Desktop, up to a maximum of 36 total files;

(iv) Place in page 2 (\$200+) and call the code necessary to delete the selected pathnames from the GS/OS Message Center; and,

(v) Launch the user's default macros as saved in 'SEG.UM', and return control to AppleWorks..

Labels

```
.FinderFileLoad
// {final release name for Ultra 4 task file portion of Init}

// Addresses

#FullPathBuffer = $800           // Work buffer with full pathname
#FileNameBuffer = $900          // Buffer with just filename
#SetUpMessage = $0200          // Start adr of MsgCntr setup routine
#ReadMessage = $020F           // Start adr of MsgCntr read routine
#ShutDownMessage = $020F       // Start adr of MsgCntr shutdown rtn
#UserID = $0203                // Assigned by MemoryManager
#MessageFlag = $0205           // Zero here = no messages
#MessageCommand = $0207        // Open ($0000) vs Print ($0001)
#MLMsgOffSet = $0209           // location in ML setting MsgOffSet

// Commands

#CursorOn = poke $10F2, 1       // cursor visible
#CursorOff = poke $10F2, 0      // cursor hidden

// Numeric Variables

#DesktopCounter = D(1)          // # of files loaded (36 max.)
#MerlinCounter = M(3)          // # of Merlin '.S' TXT files loaded
#FullPathLength = F(1)         // total length of path + filename
#FileNameLength = F(2)         // length of just filename
#JustPathLength = J(1)         // length of just path
#OffSet = O(1)                 // offset index into entire Message
#MsgBlockLW = M(1)             // Low Word of Pointer to Message
                                // for PREVIOUS filename
#MsgBlockHW = M(2)             // High Word of Pointer to Message
                                // for PREVIOUS filename
#StartByte = S(1)              // start byte of path or file name
#LastByte = L(1)               // last byte of path or file name
#LoadSuccess = L(2)            // peek $E86 ($00 = in file)
#FreeMemK = L(3)               // peekword $FD3/$FD4
#LowMem = L(4)                 // flag for low memory or load error
#MemoryManager = M(3)          // peek $FE0 ($52 = seg.rm / IIGs)
#AWFileType = A(1)             // peek $C6B ($01/02/03=ADB/AWP/ASP)
#AWPFlags = A(2)               // peek $7C68 for MM/TXT/MSDOS Flags
#AWPTextFlag = A(3)            // to store 'TXT' flag for logic (if)
#ResumeDSP = A(4)              // peekword $7C62 - pointer to last
                                // saved top display line (AW 3/4/5)

// Constants

#IsText = $20                  // $7C68 Bit 5 'on' = 'TXT' file type
                                // ($20 = 00100000)
#BeginOffSet = $02             // routine begins with offset of (2)
                                // (skips over action - open/print
                                // to first filename)
```

titles
\ba-L

Launch Default Macros

start

```
<ba-]>:<all :
poke $11AC, 0 : // disable <esc> macro abort key
$1 = screen 36,1,9 : // get to Main Menu if auto-startup

ifnot $1 = "MAIN MENU" input rtn : // is active even in cases having
rpt : // 5.25" start drive and no clock
endif :

x = peekword $B560 : // pointer to Dot command table

if x = 0 bell poke $EF70, $1E : // if pointer is '0', inits disabled
// so poke reserved <sa-ctrl-^> into
// first position of macro table
// to kill ALL macros
poke $11A2, 0 : // disable UltraMacros
oa-Q oa-S >5< rtn : // go to 'InitManager options'
msg ' You must activate Inits and reboot to use UltraMacros and this Init ' :
stop : // stop macro activity
endif :

#MemoryManager = peek $FE0: // peek $FE0 ($52 = seg.rm / IIGs)

ifnot #MemoryManager = $52 then : // ensure seg.rm (IIGs) memory mgr.
msg ' This Task file is for the Apple IIGs only ' :
.spacebar : z(1) = z : // 'Press Space Bar to continue'
ba-L : // launch default macros, plus clear
// variables and uncache this task
endif : // return - can not return

ba-B : // execute main macro functions and
:>! // then quit

<ba-[>:<all :
msg ' This Task file is to be launched from the Apple IIGs Finder only ' :
.spacebar : z(1) = z : // 'Press Space Bar to continue'
ba-L : // launch default macros, plus clear
:>! // variables and uncache this task
```

Main Routine

```
<ba-B>:<asr :
ba-5 : // poke in ML SetUp Routine @ $200
jsr #SetUpMessage : // poll MessageCenter for filenames

X(1) = peekword #MessageFlag : // Any Message available?

if X(1) = 0 then goto ba-Q : // No Message or Error - Shutdown
endif : // and quit

// ends macro IF there are NO
// files in Message Center from
// the start, and task file is
// mistakenly launched

#Offset = 0 : // initialize offset index counter
#DesktopCounter = 0 : // initialize # of files counter
#MerlinCounter = 0 : // initialize Merlin '.S' counter

ba-6 : // poke in ML Read Routine @ $20F
jsr #ReadMessage : // read/parse/store first filename

ba-1 : // parse & load first filename

begin: // loop to read more filenames

#Offset = #FullPathLength + 1 : // Jump over previous msg

#MsgBlockLW = peekword $020B : // Low Word of previous msg
#MsgBlockHW = peekword $020D : // High Word of previous msg

pokeword 4, #MsgBlockLW : // Restore Pointer to position in
pokeword 6, #MsgBlockHW : // MessageBlock containing the
// start of the previous filename
// without including any offset
// NOTE: In AUX Zero/Direct Page
// locations $01/004 - $01/007

pokeword #MLMsgOffset, #Offset : // Number of bytes to skip to jump
// over previous filename

jsr #ReadMessage : // read/parse/store next filename

ba-1 : // parse & load additional filenames

rpt : // loop until no more filenames
// (ba-1 will break out of loop if
// there are no more files and
// will then goto sa-Q to finish
// file loading)

:>!
```

Finish File Loading - show splashscreen and present Desktop Index

Q:<asr :

```
$98 = " ***   FinderFileLoader   *** " :
$97 = "- An Apple IIgs GS/OS Finder Multi-File Loader Init - " :
$96 = "       for AppleWorks 5.1       " :
$94 = "       by Hugh Hood       " :
$93 = "       version 1.3 (2024)       " :
$92 = " based on the AppleWorks 3/UltraMacros 3 'FinderLaunch' " :
$91 = " by John L. Tegelaar & Ton W. van Santen " :
$90 = "       The Netherlands (1991)       " :

$88 = "█ Total File(s) Loaded: - " + str$ #DesktopCounter + " - █" :
$84 = "┘ Merlin File(s) Loaded: - " + str$ #MerlinCounter + " - ┘" :
$83 = "Welcome to FinderFileLoader" :
$70 = %J% + "- Not all files loaded due to memory or other error -" + %K% :
```

```
.Cls 1 : // clear middle 20 lines of screen
.WriteStr 1,0,%B% : // clear top line; L0
.WriteStr 255,0,$83 : // centered; L1
.Box 255,2,60,13,1 : // draw box centered; W60; H13; MT
.WriteStr 255,4,$98 : // centered; L4
.WriteStr 255,6,$97 : // centered; L6
.WriteStr 255,7,$96 : // centered; L7
.WriteStr 255,9,$94 : // centered; L9
.WriteStr 255,10,$93 : // centered; L10
.WriteStr 255,12,$92 : // centered; L12
.WriteStr 255,13,$91 : // centered; L13
.WriteStr 255,14,$90 : // centered; L14

if #LowMem = 1 then : //
.WriteStr 255,17,$70 : //
endif : //

.WriteStr 255,18,$88 : // centered; L18
.WriteStr 255,20,$84 : // centered; L20

.spacebar : z(1) = z : // 'Press Space Bar to continue'

esc : // return to Main Menu
```

```

$82 = .PeekStr $BA32 : // read name of standard data disk
$81 = Left $82,1 :

if $81 = "/" then :
$80 = $82 :
endif :

if $82 = "RAMDisk (Slot 5)" then :
$80 = "5,2" :
endif :

$81 = Left $82,4 :

if $81 = "Disk" then :
$80 = Mid $82,14,1 + "," + Mid $82,6,1 :
endif :

.SetDisk $80 : // change disk to user's default

oa-Q : oa-1 : // present Desktop Index 1

if #DesktopCounter = 0 then :
esc : esc : // ensure back at Main Menu
endif :

ba-Q : // delete messages from MessageCenter
:>!

```

Delete Message(s) from GS/OS Message Center and Memory Manager ShutDown

```

<ba-Q>:<asr :
ba-7 : // poke in ML ShutDown Routine @ $20F
jsr #ShutDownMessage : // delete Message and ShutDown MM
ba-L : // Quit and launch default macros
:>!

```

Quit Routine - Clear Variables, Modify Launch Command & Launch Default

```

<ba-L>:<all : clear 255 : // clear all variables

$95 = "FinderFileLoad" : // final release name
// {N/A unless TimeOut menu launch}
// {seg.um will uncache $95}

```

```

// modify 'Launch' command to prevent it adding this task to macro cache
// {works by preventing cache only if current cache count = 0}

poke $FFB7, $09 :           // was $BB00 change to $900
poke $900, $A9 :           // LDA $BB
poke $901, $BB :
poke $902, $8D :           // STA $FFB7 - restore it AFTER
poke $903, $B7 :           // next Launch command is executed
poke $904, $FF :
poke $905, $A9 :           // LDA $18 - was $15
poke $906, $18 :           // skips over adding task to cache
                             // only if current cache count = 0
poke $907, $8D :           // STA $BB7F
poke $908, $7F :
poke $909, $BB :
poke $90A, $4C :           // JMP $BB00 - resume routine
poke $90B, $00 :
poke $90C, $BB :

poke $11AC, $1B :         // re-enable <esc> macro abort key
launch "seg.um" :         // Quit and launch default macros
:>!

```

Subroutine to find filename

```

<ba-1>:<asr :             // Find name & load file

X(1) = peekword #MessageFlag : // Any more Messages available?

if X(1) = 0 then goto sa-Q : // No Message or Error - finish
endif :                   // file loading

#FullPathLength = peek #FullPathBuffer : // total length of path + filename

if #FullPathLength = 0 then goto sa-Q : // No more files to load - finish
endif :                   // file loading

#FileNameLength = peek #FileNameBuffer : // length of just filename
#JustPathLength = #FullPathLength - #FileNameLength :
#JustPathLength = #JustPathLength - 1 : // exclude trailing '/'

#StartByte = #FullPathBuffer + 1 : // path starts at $801
#LastByte = #StartByte - 1 + #JustPathLength : // length inclusive of start
ba-2 : $1 = $0 : // goto Read Pathname

#StartByte = #FileNameBuffer + 1 : // filename starts at $901
#LastByte = #StartByte - 1 + #FileNameLength : // length inclusive of start
ba-2 : $2 = $0 : // goto Read Filename

ba-3 : // goto Load file
:>!

```

Get names from buffers

```
<ba-2>:<asr : // Read path/file names from buffers
$0 = "" : // initialize $0

for X(0) = #StartByte to #LastByte :
W(0) = peek X(0) :
Y(0) = .AndBits $7F, W(0) : // strip any high bits
$0 = $0 + chr$ Y(0) : // build path/file name string
next X(0) :

:>!
```

Set Path and Load file to AppleWorks Desktop

```
<ba-3>:<asr : // Load file from disk

#LowMem = 0 : // clear error message flag
#FreeMemK = peekword $FD3 : // determine K BEFORE loading
// more files

if #FreeMemK < 256 then : // NOTE: testing shows that a
#LowMem = 1 : // IIGs with less than
sa-Q : // 256K remaining can
endif : // encounter an out-of-
// memory error on larger
// files and can crash
// when under macro control
// in spite of methods to
// trap the condition and
// react accordingly

if #DesktopCounter = 0 then :
.filecard "Main Menu", 1 : // build 1st level filecard
.filecard "Add Files", 2 : // build 2nd level filecard
.filecard "", 3 : // build blank 3rd level filecard
.writestr 50, 6, "File(s) Loaded: " : // present # of files message
$89 = "Loading the file - one moment" :
.writestr 255, 12, $89 : // present activity message
.writestr 255, 14, "┆" : // present hourglass icon
endif :
```



```

display #off : // hide steps to get to Main Menu
.therm 22, 17, 0, 100 : // initialize thermometer (0%)
$87 = " " : // message blanking string
.writestr 255, 0, $87 :
.writestr 14, 7, $87 :
.writestr 255, 16, $87 :
.writestr 0, 0, "↑: " + $1 : // show loading path in upper left
.writestr 255, 0, ''' + $2 + ''' :
.writestr 14, 7, ''' + $2 + ''' :
.writestr 255, 16, $2 : // present filename message
.writestr 66, 6, str$ #DesktopCounter :
oa-Q : esc : // Main Menu for sure (for .SetDisk)
.SetDisk $1 : // change to pathname for next file
rtn : // select 'Add files..'
// DIRECT 'Enter name:' Method
.therm 22, 17, 50, 100 : // show 50% progress thermometer
poke $2271, $60 : // bypass 'more than one' question
oa-rtn : // 'Enter name'

poke $FD1, $01 : // suppress DT Full Warning message

print $2 : rtn : // type filename and load
.therm 22, 17, 100, 100 : // show 100% progress thermometer
#LoadSuccess = peek $E86 : // 0=in file / 1=menu / 64= list

ifnot #LoadSuccess = 0 then : // finish if file load not successful

esc : esc : // ensure back at Main Menu in case
// of Desktop Full condition

#LowMem = 1 : // set error message flag

| : esc : // update underlying screen before
// re-enabling screen
// {'|' command is 'BAR'}

display #on : // (re-enable screen first)
poke $11AC, 0 : // re-disable <esc> macro abort key

poke $FD1, 0 : // Restore Desktop Full Warning

sa-Q : // finish file loading
endif : //

$85 = right $2,2 : // last (2) characters of filename?
#AWPFlags = peek $7C68 // MM/TXT/MSDOS Flags here
#AWPTextFlag = .AndBits #AWPFlags,#IsText : // read 'TXT' bit5 in #AWPFlags
#AWFileType = peek $C6B // $01/$02/$03 = ADB/AWP/ASP

if $85 = ".S" or $85 = ".s" and : // is a Merlin Source File suffix?
#AWFileType = 2 and : // are we in the AWP Seg?
#AWPTextFlag = #IsText : // is 'TXT' (not 'AWP') file type?
then ba-4 : // set Tab Ruler for Merlin file
endif : //

#ResumeDSP = peekword $7C62 // pointer to last saved top
// display line (AW3/4/5)

```

```

if #AWFileType = 2 and : // are we in the AWP Seg?
#ResumeDSP > $7D37 then : // Loaded AppleWorks 2.x AWP File
oa-1 : // so go to top of file since
endif : // cursor location is not valid

```

```

#DesktopCounter = #DesktopCounter + 1 : // keep track of # of files loaded

```

```

if #DesktopCounter > 36 then : // Quit if (3) desktops become full
display #on : // (re-enable screen first)
poke $11AC, 0 : // re-disable <esc> macro abort key
sa-Q: // finish file loading
endif :

```

```

display #on : // re-enable screen (don't refresh)
poke $11AC, 0 : // re-disable <esc> macro abort key
:>!

```

Set Screen Tabs / Tab Ruler for Merlin Source File

```

<ba-4>:<asr : //
    $86 =
"===== :
// Merlin Source File Tab Ruler

.PokeStr $86, $7C10 : // insert Merlin Tab Ruler in file
oa-Q : rtn : // make changes 'stick'
oa-T : // open Tab Ruler menu
>m< esc : // select 'm'odify current & exit
#MerlinCounter = #MerlinCounter + 1 : // keep track of Merlin files loaded
:>!

```

Poke in Machine Language Routine to Poll GS/OS Message Center - SetUp

** For Merlin source code for this routine see "AWLOADERSU.S" **

```

<ba-5>:<asr : // poke SetUp ML routine
poke $0200, $4C : // Jump to $020F
poke $0201, $0F : // "
poke $0202, $02 : // "

poke $0203, $00 : // UserID
poke $0204, $00 : // "
poke $0205, $00 : // MessageFlag
poke $0206, $00 : // "
poke $0207, $00 : // MessageCommand

```

```
poke $0208, $00 : // "  
poke $0209, $02 : // MLMsgOffset  
poke $020A, $00 : // "  
poke $020B, $00 : // MsgBlockLW  
poke $020C, $00 : // "  
poke $020D, $00 : // MsgBlockHW  
poke $020E, $00 : // "  
  
poke $020F, $18 : // SetUp ML routine real start  
poke $0210, $FB :  
poke $0211, $C2 :  
poke $0212, $30 :  
poke $0213, $48 :  
poke $0214, $A2 :  
poke $0215, $02 :  
poke $0216, $02 :  
poke $0217, $22 :  
  
poke $0218, $00 :  
poke $0219, $00 :  
poke $021A, $E1 :  
poke $021B, $68 :  
poke $021C, $8D :  
poke $021D, $03 :  
poke $021E, $02 :  
poke $021F, $48 :  
  
poke $0220, $48 :  
poke $0221, $F4 :  
poke $0222, $00 :  
poke $0223, $00 :  
poke $0224, $F4 :  
poke $0225, $80 :  
poke $0226, $00 :  
poke $0227, $AD :  
  
poke $0228, $03 :  
poke $0229, $02 :  
poke $022A, $48 :  
poke $022B, $F4 :  
poke $022C, $1C :  
poke $022D, $00 :  
poke $022E, $F4 :  
poke $022F, $00 :  
  
poke $0230, $00 :  
poke $0231, $F4 :  
poke $0232, $00 :  
poke $0233, $00 :  
poke $0234, $A2 :  
poke $0235, $02 :  
poke $0236, $09 :  
poke $0237, $22 :
```

poke \$0238, \$00 :
poke \$0239, \$00 :
poke \$023A, \$E1 :
poke \$023B, \$68 :
poke \$023C, \$85 :
poke \$023D, \$00 :
poke \$023E, \$68 :
poke \$023F, \$85 :

poke \$0240, \$02 :
poke \$0241, \$F4 :
poke \$0242, \$02 :
poke \$0243, \$00 :
poke \$0244, \$F4 :
poke \$0245, \$01 :
poke \$0246, \$00 :
poke \$0247, \$A5 :

poke \$0248, \$02 :
poke \$0249, \$48 :
poke \$024A, \$A5 :
poke \$024B, \$00 :
poke \$024C, \$48 :
poke \$024D, \$A2 :
poke \$024E, \$01 :
poke \$024F, \$15 :

poke \$0250, \$22 :
poke \$0251, \$00 :
poke \$0252, \$00 :
poke \$0253, \$E1 :
poke \$0254, \$90 :
poke \$0255, \$06 :
poke \$0256, \$A9 :
poke \$0257, \$00 :

poke \$0258, \$00 :
poke \$0259, \$82 :
poke \$025A, \$1A :
poke \$025B, \$00 :
poke \$025C, \$A7 :
poke \$025D, \$00 :
poke \$025E, \$18 :
poke \$025F, \$69 :

poke \$0260, \$06 :
poke \$0261, \$00 :
poke \$0262, \$85 :
poke \$0263, \$04 :
poke \$0264, \$A0 :
poke \$0265, \$02 :
poke \$0266, \$00 :
poke \$0267, \$B7 :

poke \$0268, \$00 :
poke \$0269, \$69 :
poke \$026A, \$00 :
poke \$026B, \$00 :
poke \$026C, \$85 :
poke \$026D, \$06 :
poke \$026E, \$A7 :
poke \$026F, \$04 :

poke \$0270, \$8D :
poke \$0271, \$07 :
poke \$0272, \$02 :
poke \$0273, \$A9 :
poke \$0274, \$FF :
poke \$0275, \$FF :
poke \$0276, \$8D :
poke \$0277, \$05 :

poke \$0278, \$02 :
poke \$0279, \$20 :
poke \$027A, \$83 :
poke \$027B, \$02 :
poke \$027C, \$EA :
poke \$027D, \$EA :
poke \$027E, \$38 :
poke \$027F, \$FB :

poke \$0280, \$E2 :
poke \$0281, \$30 :
poke \$0282, \$60 :
poke \$0283, \$A5 :
poke \$0284, \$04 :
poke \$0285, \$8D :
poke \$0286, \$0B :
poke \$0287, \$02 :

poke \$0288, \$A5 :
poke \$0289, \$06 :
poke \$028A, \$8D :
poke \$028B, \$0D :
poke \$028C, \$02 :
poke \$028D, \$60 :
:>!

Poke in Machine Language Routine to Read Filenames in GS/OS Message Center

** For Merlin source code for this routine see "AWLOADERRD.S" **

```
<ba-6>:<asr : // poke Read ML routine
poke $020F, $18 :
poke $0210, $FB :
poke $0211, $C2 :
poke $0212, $30 :
poke $0213, $18 :
poke $0214, $AD :
poke $0215, $09 :
poke $0216, $02 :
poke $0217, $65 :

poke $0218, $04 :
poke $0219, $85 :
poke $021A, $04 :
poke $021B, $A9 :
poke $021C, $00 :
poke $021D, $00 :
poke $021E, $65 :
poke $021F, $06 :

poke $0220, $85 :
poke $0221, $06 :
poke $0222, $A0 :
poke $0223, $00 :
poke $0224, $00 :
poke $0225, $B7 :
poke $0226, $04 :
poke $0227, $99 :

poke $0228, $00 :
poke $0229, $08 :
poke $022A, $C8 :
poke $022B, $C8 :
poke $022C, $C0 :
poke $022D, $80 :
poke $022E, $00 :
poke $022F, $90 :

poke $0230, $F4 :
poke $0231, $E2 :
poke $0232, $30 :
poke $0233, $AD :
poke $0234, $00 :
poke $0235, $08 :
poke $0236, $F0 :
poke $0237, $30 :
```

poke \$0238, \$48 :
poke \$0239, \$A8 :
poke \$023A, \$B9 :
poke \$023B, \$00 :
poke \$023C, \$08 :
poke \$023D, \$88 :
poke \$023E, \$C9 :
poke \$023F, \$2F :

poke \$0240, \$D0 :
poke \$0241, \$F8 :
poke \$0242, \$98 :
poke \$0243, \$8D :
poke \$0244, \$00 :
poke \$0245, \$08 :
poke \$0246, \$C8 :
poke \$0247, \$C8 :

poke \$0248, \$8C :
poke \$0249, \$00 :
poke \$024A, \$09 :
poke \$024B, \$68 :
poke \$024C, \$8D :
poke \$024D, \$00 :
poke \$024E, \$08 :
poke \$024F, \$1A :

poke \$0250, \$38 :
poke \$0251, \$ED :
poke \$0252, \$00 :
poke \$0253, \$09 :
poke \$0254, \$AC :
poke \$0255, \$00 :
poke \$0256, \$09 :
poke \$0257, \$8D :

poke \$0258, \$00 :
poke \$0259, \$09 :
poke \$025A, \$A2 :
poke \$025B, \$01 :
poke \$025C, \$B9 :

poke \$025D, \$00 :
poke \$025E, \$08 :
poke \$025F, \$9D :

poke \$0260, \$00 :
poke \$0261, \$09 :
poke \$0262, \$E8 :
poke \$0263, \$C8 :
poke \$0264, \$E0 :
poke \$0265, \$10 :
poke \$0266, \$90 :
poke \$0267, \$F4 :

```
poke $0268, $C2 :
poke $0269, $30 :
poke $026A, $A9 :
poke $026B, $FF :
poke $026C, $FF :
poke $026D, $8D :
poke $026E, $05 :
poke $026F, $02 :
```

```
poke $0270, $20 :
poke $0271, $7A :
poke $0272, $02 :
poke $0273, $EA :
poke $0274, $EA :
poke $0275, $38 :
poke $0276, $FB :
poke $0277, $E2 :
```

```
poke $0278, $30 :
poke $0279, $60 :
poke $027A, $A5 :
poke $027B, $04 :
poke $027C, $8D :
poke $027D, $0B :
poke $027E, $02 :
poke $027F, $A5 :
```

```
poke $0280, $06 :
poke $0281, $8D :
poke $0282, $0D :
poke $0283, $02 :
poke $0284, $60 :
:>!
```

Poke in Machine Language Routine to Shutdown GS/OS Message Center - ShutDown

** For Merlin source code for this routine see "AWLOADERSD.S" **

```
<ba-7>:<a11 : // poke ShutDown ML routine
poke $020F, $18 :
poke $0210, $FB :
poke $0211, $C2 :
poke $0212, $30 :
poke $0213, $AD :
poke $0214, $03 :
poke $0215, $02 :
poke $0216, $48 :
poke $0217, $A5 :
```


poke \$0218, \$02 :
poke \$0219, \$48 :
poke \$021A, \$A5 :
poke \$021B, \$00 :
poke \$021C, \$48 :
poke \$021D, \$F4 :
poke \$021E, \$03 :
poke \$021F, \$00 :

poke \$0220, \$F4 :
poke \$0221, \$01 :
poke \$0222, \$00 :
poke \$0223, \$48 :
poke \$0224, \$48 :
poke \$0225, \$A2 :
poke \$0226, \$01 :
poke \$0227, \$15 :

poke \$0228, \$22 :
poke \$0229, \$00 :
poke \$022A, \$00 :
poke \$022B, \$E1 :
poke \$022C, \$A2 :
poke \$022D, \$02 :
poke \$022E, \$10 :
poke \$022F, \$22 :

poke \$0230, \$00 :
poke \$0231, \$00 :
poke \$0232, \$E1 :
poke \$0233, \$A2 :
poke \$0234, \$02 :
poke \$0235, \$03 :
poke \$0236, \$22 :
poke \$0237, \$00 :

poke \$0238, \$00 :
poke \$0239, \$E1 :
poke \$023A, \$38 :
poke \$023B, \$FB :
poke \$023C, \$E2 :
poke \$023D, \$30 :
poke \$023E, \$60 :
:>!

end