```
**********************
 1
 2
 3
   *
                    AWLOADERSU Merlin Source File
 4
                         - SetUp (SU) File -
 5
            See Also Read (RD) File and ShutDown (SD) Files
 6
 7
                  (For use with FinderFileLoad Init)
 8
 9
                             by Hugh Hood
10
                          Version 1.3 (2024)
11
12
13
        based on the AppleWorks 3/UltraMacros 'FinderLaunch'
             by John L. Tegelaar and Ton W. van Santen
14
15
                      The Netherlands (1991)
16
17
18
19
               TR
                           ADR
                                        ; truncate bank address
20
21
               XC
                                        ; enable 65C02 code
22
               XC
                                        : enable 65C816 code
23
24
               ORG
                            $0200
                                        ; AppleWorks Load Location
25
               TYP
                            $06
                                        ; create binary file
26
27 FullBuff
               EQU
                            $800
                                        ; fullpath work buffer
28
   FileBuff
               EQU
                            $900
                                        ; filename work buffer
29
30
31
               JMP
                            Begin
32 UserID
                            $0000
                                        ; UserID assigned by Memory Manager
               DA
33
   MsqFlaq
               DA
                            $0000
                                         Initial value is #$0000 (no message)
                                         Open ($0000) vs. Print ($0001)
34 MsqCmnd
               DA
                            $0000
35
                                           (both types are loaded to AppleWorks)
36 MsgOffset
               DA
                            $0002
                                          Offset into Message Block for filenames
                                           (initial value is #$0002 for File #1)
37
                                          Reading Address including Offsets
38 MsqBlock
               ADRL
                            $00000000
39
40
   Begin
41
   Mode16
               CLC
                                          prep for native mode
42
               XCE
                                         now in native mode
43
               REP
                           #$30
                                          16-bit registers (X,Y,A)
44
45
   Setup
               PHA
                                        ; Push (2) Bytes on Stack as space
46
                                          UserID result
47
                           #$0202
                                         Memory Manager Startup
               LDX
48
               JSL
                            $E10000
                                         Tool Loader Routine
49
               PLA
                                         Pull UserID from Stack
50
               STA
                           UserID
                                        ; Store UserID
```

PHA SPEA SPEA SPEA SPEA SPEA SPEA SPEA SPE				
PLA   \$0000   Push (4) Bytes on Stack Specifying   Size of Memory Block Required   Size of Memory Block Required   Size of Memory Block Required   Push UserID   Push UserID   Push UserID on Stack   Push Memory Attribute Word on Stack   Required   Push Memory Attribute Word on Stack   Push Memory Attribute Word on Stack   Required   Push (4) Bytes on Stack Specifying   Push (4) Bytes on Stack Required (N/A)   Push Low Handle from Stack   Push Specify Bytes   Push (4) Bytes on Stack Required (N/A)   Push Word of Handle from Stack   Push Specify Bytes   Push Action #2 (Get Message) on Stack   Push Specify Bytes   Push Stack Pesh Specify Bytes   Push Specify Bytes   Push Stack   Push Specify Bytes   Push Specify Bytes   Push Stack   Push Specify Bytes   Push Specify Bytes   Push Stack   Push Specify Bytes   Push Spec		PHA		; Push (4) Bytes on Stack as space for
PEA		PHA		; Handle Result
DA		PEA	·	
PHA				
PEA		LDA	UserID	,
\$   (no special - page aligned - no cross)  PEA		PHA		,
S9 PEA \$0000 ; Push (4) Bytes on Stack Specifying 60 PEA \$0000 ; Location of Memory Block Required (N/A) 61 LDX #\$9002 ; Request a New Handle 62 JSL \$E10000 ; Tool Loader Routine ; Pull Low Word of Handle at \$00/\$01 E55 PLA ; Pull High Word of Handle at \$00/\$01 PlA ; Pull High Word of Handle at \$00/\$01 PlA ; Pull High Word of Handle at \$02/\$03 PlA ; Store High Word of Handle at \$02/\$03 PlA ; Push Action #2 (Get Message) on Stack 66 STA \$000 ; Push Action #2 (Get Message) on Stack 67 PEA \$0001 ; Push Type/ID #1 (Filename) on Stack 68 PEA \$0001 ; Push Type/ID #1 (Filename) on Stack 69 LDA \$02 ; Load High Word of Handle from Stack 70 PHA ; Push High Word of Handle on Stack 70 PHA ; Push Low Word of Handle on Stack 71 LDA \$00 ; Load Low Word of Handle on Stack 71 LDA \$00 ; Load Low Word of Handle on Stack 72 PHA ; Push Low Word of Handle on Stack 73 LDX #\$1501 ; Retrieve From MessageCenter 74 JSL \$E10000 ; Tool Loader Routine 75 BCC NoError ; Proceed if No Error (carry clear) 76 LDA #\$0000 ; If error, prepare MsgFlag to Receive \$0000 PRE From Proceed to Exit 78 Proceed to Exit 79 NoError LDA [\$00] ; Load Low Word of Message Block Address 70 Prepare for Addition 70 Add (6) Bytes (offset) to Low Word 71 Add (6) Bytes (offset) to Low Word 71 Address 71 Address 72 Address 73 Add (7) Store Low Word of Message Block Address 74 Address 75 Add		PEA	\$001C	; Push Memory Attribute Word on Stack
60				
61				
SEL				
PLA   STA   \$00   \$ Store Low Word of Handle from Stack   \$10				
STA			\$E10000	
Full High Word of Handle from Stack				,
STA   \$02   Store High Word of Handle at \$02/\$03			\$00	•
PEA				,
PEA				,
69 LDA \$02 ; Load High Word of Handle from Stack 70 PHA ; Push High Word of Handle on Stack 71 LDA \$00 ; Load Low Word of Handle on Stack 71 Push Low Word of Handle on Stack 72 PHA ; Push Low Word of Handle on Stack 73 LDX #\$1501 ; Retrieve From MessageCenter 74 JSL \$E10000 ; Tool Loader Routine 75 BCC NoError ; Proceed if No Error (carry clear) 76 LDA #\$0000 ; If error, prepare MsgFlag to Receive \$0000 77 BRL Exit ; Proceed to Exit 78 NoError LDA [\$00] ; Load Low Word of Message Block Address 79 Referenced by Handle at \$00/\$01/\$02/\$03 ; Prepare for Addition 79 Referenced by Handle at \$00/\$01/\$02/\$03 ; Prepare for Addition 79 Referenced 80 Ressage Block Address 70 Referenced 80 Ressage Block Address 80 Referenced 80 Ressage Block Address 80 Referenced 80 Ressage 80 Referenced 80 Referenced 80 Ressage 80 Referenced 80 Ressage 80 Referenced 80 Referenced 80 Referenced 80 Ressage 80 Referenced 80 Ressage 80 Referenced 80 Re				,
PHA				
Till			\$02	,
PHA			<b>#</b> 00	,
Tool Loader Routine			\$00	,
Tool Loader Routine   Tool Loader Routine			U#4 F04	
RCC NoError ; Proceed if No Error (carry clear)  RDA #\$0000 ; If error, prepare MsgFlag to Receive \$0000  RRL Exit ; Proceed to Exit  NoError LDA [\$00] ; Load Low Word of Message Block Address  Referenced by Handle at \$00/\$01/\$02/\$03  CLC ; Prepare for Addition  ADC #\$0006 ; Add (6) Bytes (offset) to Low Word  ADC #\$0006 ; Add (6) Bytes (offset) to Low Word  ADC #\$0002 ; Prepare 'Y' Index to Read High Word  LDA [\$00],Y ; Load High Word of Message Block Address  Referenced by Handle at \$00/\$01/\$02/\$03  ADC #\$0000; Prepare 'Y' Index to Read High Word  LDA [\$00],Y ; Load High Word of Message Block Address  Referenced by Handle at \$00/\$01/\$02/\$03  ADC #\$0000 ; Add (0) Bytes (no offset) to High Word  Address at \$06/07  LDA [\$04] ; Load Low Word from Message Block  Referenced by Address at \$04/\$05/\$06/\$07  STA MsgCmnd ; Store Low Word at MsgCmnd -  Open (\$0000) vs. Print (\$0001)  (both types are loaded to AppleWorks)  AMCRO Loops for next file (if present)				
#\$0000 ; If error, prepare MsgFlag to Receive \$0000   RRL				•
Total Color				
78 79 NoError LDA [\$00] ; Load Low Word of Message Block Address ; Referenced by Handle at \$00/\$01/\$02/\$03 81 CLC ; Prepare for Addition 82 ADC #\$0006 ; Add (6) Bytes (offset) to Low Word 83 84 STA \$04 ; Store Low Word of Message Block 85 LDY #\$0002 ; Prepare 'Y' Index to Read High Word 87 LDA [\$00],Y ; Load High Word of Message Block Address 88 Referenced by Handle at \$00/\$01/\$02/\$03 89 ADC #\$0000 ; Add (0) Bytes (no offset) to High Word 90 STA \$06 ; Store High Word of Message Block 91 2 LDA [\$04] ; Load Low Word from Message Block 93 STA MsgCmnd ; Store Low Word at MsgCmnd - 35 95 96 10 10 10 10 10 10 10 10 10 10 10 10 10				
NoError LDA [\$00] ; Load Low Word of Message Block Address ; Referenced by Handle at \$00/\$01/\$02/\$03   81		BKL	EXIT	; Proceed to Exit
Referenced by Handle at \$00/\$01/\$02/\$03  CLC		No Francia I DA	Γάρο	Load Low World of Mossage Diock Address
81 CLC  82 ADC #\$0006 ; Add (6) Bytes (offset) to Low Word  83 ; of Message Block Address  84 STA \$04 ; Store Low Word of Message Block  85 ; Address + Offset at \$04/\$05  86 LDY #\$0002 ; Prepare 'Y' Index to Read High Word  87 LDA [\$00],Y ; Load High Word of Message Block Address  88 ; Referenced by Handle at \$00/\$01/\$02/\$03  89 ADC #\$0000 ; Add (0) Bytes (no offset) to High Word  90 STA \$06 ; Store High Word of Message Block  91 ; Address at \$06/07  92 LDA [\$04] ; Load Low Word from Message Block  93 ; Referenced by Address at \$04/\$05/\$06/\$07  94 STA MsgCmnd ; Store Low Word at MsgCmnd -  95 ; Open (\$0000) vs. Print (\$0001)  96 ; (both types are loaded to AppleWorks)  97 LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure  98 ; MACRO Loops for next file (if present)		NOETTOT LDA	ראַשע	,
#\$0006		CLC		
### STA ### \$\text{\$04}\$   \$\text{\$04}\$   \$\text{\$05}\$   \$\text{\$04}\$   \$\text{\$06}\$   \$\text{\$06}\$   \$\text{\$06}\$   \$\text{\$06}\$   \$\text{\$07}\$   \$\$0			#\$0006	·
\$4 STA \$04 ; Store Low Word of Message Block  \$5		ADC	#\$0000	
35		STA	\$0A	,
#\$0002 ; Prepare 'Y' Index to Read High Word  LDA [\$00],Y ; Load High Word of Message Block Address  Referenced by Handle at \$00/\$01/\$02/\$03  ADC #\$0000 ; Add (0) Bytes (no offset) to High Word  STA \$06 ; Store High Word of Message Block  Address at \$06/07  LDA [\$04] ; Load Low Word from Message Block  Referenced by Address at \$04/\$05/\$06/\$07  STA MsgCmnd ; Store Low Word at MsgCmnd -  Open (\$0000) vs. Print (\$0001)  (both types are loaded to AppleWorks)  LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure  MACRO Loops for next file (if present)		SIA	<b>₽</b> VT	,
LDA		I DY	#\$0002	
Referenced by Handle at \$00/\$01/\$02/\$03  ADC #\$0000 ; Add (0) Bytes (no offset) to High Word  STA \$06 ; Store High Word of Message Block  Address at \$06/07  LDA [\$04] ; Load Low Word from Message Block  Referenced by Address at \$04/\$05/\$06/\$07  STA MsgCmnd ; Store Low Word at MsgCmnd -  Open (\$0000) vs. Print (\$0001)  (both types are loaded to AppleWorks)  LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure  MACRO Loops for next file (if present)				, ,
ADC #\$0000 ; Add (0) Bytes (no offset) to High Word  STA \$06 ; Store High Word of Message Block  Address at \$06/07  LDA [\$04] ; Load Low Word from Message Block  Referenced by Address at \$04/\$05/\$06/\$07  STA MsgCmnd ; Store Low Word at MsgCmnd -  Open (\$0000) vs. Print (\$0001)  Open (\$0000) vs. Print (\$0001)  Store Low Word at MsgCmnd -  Open (\$0000) vs. Print (\$0001)  MACRO Loops for next file (if present)		LDA	[400],1	, and the second
90 STA \$06 ; Store High Word of Message Block 91 ; Address at \$06/07  92 LDA [\$04] ; Load Low Word from Message Block 93 ; Referenced by Address at \$04/\$05/\$06/\$07  94 STA MsgCmnd ; Store Low Word at MsgCmnd - 95 ; Open (\$0000) vs. Print (\$0001) 96 ; (both types are loaded to AppleWorks)  97 LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure 98 ; MACRO Loops for next file (if present)		ADC	#\$0000	
; Address at \$06/07  102				
92 LDA [\$04] ; Load Low Word from Message Block 93 ; Referenced by Address at \$04/\$05/\$06/\$07 94 STA MsgCmnd ; Store Low Word at MsgCmnd - 95 ; Open (\$0000) vs. Print (\$0001) 96 ; (both types are loaded to AppleWorks) 97 LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure 98 ; MACRO Loops for next file (if present)			7	,
; Referenced by Address at \$04/\$05/\$06/\$07  STA MsgCmnd; Store Low Word at MsgCmnd - ; Open (\$0000) vs. Print (\$0001) ; (both types are loaded to AppleWorks)  LDA #\$FFFF; Set MsgFlag to non-zero value to Ensure ; MACRO Loops for next file (if present)		LDA	Γ\$047	
94 STA MsgCmnd; Store Low Word at MsgCmnd - 95 ; Open (\$0000) vs. Print (\$0001) 96 ; (both types are loaded to AppleWorks) 97 LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure 98 ; MACRO Loops for next file (if present) 99				,
; Open (\$0000) vs. Print (\$0001) ; (both types are loaded to AppleWorks) 97 LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure 98 ; MACRO Loops for next file (if present) 99		STA	MsqCmnd	
; (both types are loaded to AppleWorks)  Property LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure  MACRO Loops for next file (if present)  MACRO Loops for next file (if present)			<b>3</b> · ·	
97 LDA #\$FFFF ; Set MsgFlag to non-zero value to Ensure 98 ; MACRO Loops for next file (if present) 99	96			
98 ; MACRO Loops for next file (if present) 99		LDA	#\$FFFF	
99				,
100 Exit STA MsgFlag ; Store either #\$0000 (no message) or				
	100	Exit STA	MsgFlag	; Store either #\$0000 (no message) or

101				; #\$FFFF (message) at MsgFlag for MACRO
102		JSR	SavOffset	; Save offset(s) into Message Block for
103				; CURRENT file's full name
104		NOP		; Spare Byte
105		NOP		; Spare Byte
106				, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
	Mode8	SEC		; prep for emulation mode
108	riodeo	XCE		; now in emulation mode
109		SEP	#\$30	; 8-bit registers (X,Y,A)
110		JEI	11 \$30	, o bit registers (X,1,A)
	Done	RTS		; return to MACRO calling routine
112	Done	KIS		, recuir to macke calling reactine
	SavOffset	LDA	\$04	; Load Low Word of CURRENT file's Message
114	Javorrsec	LDA	<b>₽</b> ₩	; Block Address including Offsets
115		STA	MsgBlock	; Store Low Word of CURRENT file's Message
116		SIA	MSGBLOCK	; Block Address including Offsets as a
117				; base address to add offset for next
118				,
-		LDA	¢oc.	; file's reading
119		LDA	\$06	; Load High Word of CURRENT file's Message
120		CTA	W D1 1 2	; Block Address including Offsets (none)
121		STA	MsgBlock+2	; Store High Word of CURRENT file's Message
122		D.T.C		; Block Address including Offsets (none)
123		RTS		; Return to Exit routine
124				
125	End			;
126				
127		SAV	AWLOADERSU.	OBJ
128				

```
**********************
 1
 2
 3
   *
                    AWLOADERRD Merlin Source File
 4
                         - Read (RD) File -
 5
          See Also SetUp (SU) File and ShutDown (SD) File
 6
 7
                  (For use with FinderFileLoad Init)
 8
 9
                             by Hugh Hood
10
                          Version 1.3 (2024)
11
12
13
        based on the AppleWorks 3/UltraMacros 'FinderLaunch'
             by John L. Tegelaar and Ton W. van Santen
14
15
                      The Netherlands (1991)
16
17
18
19
               TR
                           ADR
                                        ; truncate bank address
20
21
               XC
                                         enable 65C02 code
22
               XC
                                         enable 65C816 code
23
24
               ORG
                           $020F
                                        ; AppleWorks Load Location
25
               TYP
                           $06
                                        ; create binary file
26
27 UserID
               EQU
                           $203
                                        ; UserID assigned by Memory Manager
28
   MsqFlaq
               EQU
                           $205
                                         Initial value is #$0000 (no message)
29
   MsqCmnd
               EQU
                           $207
                                        ; Open ($0000) vs. Print ($0001)
30
                                          (both types are loaded to AppleWorks)
31 MsgOffset
               EQU
                           $209
                                        ; Offset into Message Block for filenames
32
                                          (initial value is #$0002 for File #1)
33 MsqBlock
               EQU
                           $20B
                                        ; Reading Address including Offsets
34
35 FullBuff
               EQU
                           $800
                                        ; fullpath work buffer
36 FileBuff
               EQU
                           $900
                                        ; filename work buffer
37
38
39 Begin
40
   Mode16
               CLC
                                          prep for native mode
41
               XCE
                                         now in native mode
42
               REP
                           #$30
                                         16-bit registers (X,Y,A)
43
44
45
               CLC
                                        ; Prepare for Addition
46
               LDA
                           MsgOffset
                                         Increase Offset into Low Word of
47
               ADC
                           $04
                                          Message Block Address by (xxxx) Bytes
48
                                           NOTE: initial value is #$0002 and
                                                  is dynamically increased during
49
                                                  the execution of MACRO Task file
50
```

51				to mood and load multiple files
52				<pre> ; to read and load multiple files ; passed in the same Message by </pre>
53				; adding PREVIOUS file's length
54				; + \$01 byte to PREVIOUS Offset
55		STA	\$04	
56		STA	<b>304</b>	; Store Low Word of Message Block Address ; + (\$06) + (\$xxxx) Offsets at \$04/\$05
57		LDA	#¢0000	
58			#\$0000	; Add (0) Bytes (no offset) to High Word
59		ADC	\$06	; of Message Block Address at \$06/\$07
		STA	\$06	; Store High Word of Message Block
60 61		LDY	#¢0000	; Address + (\$00) Offset at \$06/\$07
62	Poadl oon		#\$0000	; Initialize fullpath ReadLoop Index counter
63	ReadLoop	LDA	[\$04],Y	; Load fullpath word-by-word (Index 'Y')
64				; from Message Block Referenced by ; Address at \$04/\$05/\$06/\$07
65		STA	FullBuff,Y	; Store fullpath word-by-word (Index 'Y')
66		STA	rullbuii,i	; in fullpath work buffer at \$200
67		INY		; Increment Index 'Y' counter by 1 byte
68		INY		; twice (1 word total)
69		CPY	#\$0080	; Compare byte count with 128
70		CFT	# \$0000	; (ProDOS prefix + partial pathname limit)
71		ВСС	ReadLoop	; Continue with fullpath ReadLoop if counter
72		ВСС	Reductoop	; is less than #\$0080 (128) Bytes
73				, 13 less than #\$0000 (120) bytes
74		SEP	#\$30	; 8-bit registers (X,Y,A) / single bytes
75		LDA	FullBuff	; Load fullpath length byte from start
76		LDA	Tarrbatt	; of fullpath work buffer at \$200
77		BEQ	Finish	; If fullpath length byte is \$00 then
78		DLQ	1 1111311	; there are No More Files - goto Finish
79		PHA		; Push fullpath length byte on Stack
80		THA		; for restoring later
81				, Tol Testoring Later
82		TAY		; Copy fullpath length byte to Index 'Y'
83	ScanLoop	LDA	FullBuff,Y	; Load fullpath byte-by-byte from end
84		DEY		; Decrement Index 'Y' counter by 1 byte
85		CMP	#\$2F	; Compare byte to '/' (path separator)
86		BNE	ScanLoop	; Continue backward read until '/' is found
87			<b></b>	,
88		TYA		; Put length of path (only) in accumulator
89		STA	FullBuff	; Temporarily store length of path (only)
90				; at start of fullpath work buffer at \$200
91		INY		; Increment 'Y' [path (only)] by (1) byte
92				; (to move to leading '/' position)
93		INY		; Increment 'Y' again by (1) byte
94				; (to move to first character of filename)
95		STY	FileBuff	; Temporarily store path (only) length
96				; + (2) bytes at start of filename
97				; work buffer at \$800
98				
99		PLA		; Pull fullpath length byte back from Stack
100		STA	FullBuff	; Restore fullpath length byte at start

	I			
101				; of fullpath work buffer at \$200
102		TNC		T
103		INC		; Increment Accumulator 'A' by (1) byte
104 105		SEC	r:1 opues	; Prepare for Subtraction
106		SBC	FileBuff	; Subtract path (only) length + (2) bytes
107				<pre>; from the Accumulator (fullpath length ; + (1) byte) leaving filename length</pre>
108				; in the Accumulator
109		LDY	FileBuff	; Load 'Y' with path (only) length
110		LUI	TELEBUTT	; + (2) bytes
111		STA	FileBuff	; Store filename (only) length at start
112		317.	reebarr	; of filename work buffer at \$800
113		LDX	#\$01	; Initialize filename (only) StoreFN Loop
114			40=	; Index Counter beginning at #\$01
	StoreFN	LDA	FullBuff,Y	,
116			,	; at end of path (only) / start of file
117		STA	FileBuff,X	; Store filename byte-by-byte beginning
118				; at FileBuff (\$800) + (1) byte
119		INX		; Increment Index 'X' by (1) byte
120		INY		; Increment Index 'Y' by (1) byte
121		CPX	#\$10	; Ensure Max of 15 characters (ProDOS)
122		BCC	StoreFN	; Continue read & store operation up to
123				; ProDOS maximum of 15 character filename
124				
	Finish	REP	#\$30	; 16-bit registers (X,Y,A)
126		LDA	#\$FFFF	; Set MsgFlag to non-zero value to Ensure
127				; MACRO Loops for next file (if present)
128	F	CTA	МГ1	(Classes dillara ##0000 (cas masses)
	Exit	STA	MsgFlag	; Store either #\$0000 (no message) or
130 131		1CD	SavOffset	; #\$FFFF (message) at MsgFlag for MACRO
132		JSR	Savorrset	<pre>; Save offset(s) into Message Block for ; CURRENT file's full name</pre>
133		NOP		; Spare Byte
134		NOP		; Spare Byte
135				, 55%, 6 5, 66
	Mode8	SEC		; prep for emulation mode
137		XCE		; now in emulation mode
138		SEP	#\$30	; 8-bit registers (X,Y,A)
139				
140	Done	RTS		; return to MACRO calling routine
141				
142	SavOffset	LDA	\$04	; Load Low Word of CURRENT file's Message
143				; Block Address including Offsets
144		STA	MsgBlock	; Store Low Word of CURRENT file's Message
145				; Block Address including Offsets as a
146				; base address to add offset for next
147		LDA	¢oc.	; file's reading
148		LDA	\$06	; Load High Word of CURRENT file's Message
149		СТА	McaDl calc. 2	; Block Address including Offsets (none)
150		STA	MsgBlock+2	; Store High Word of CURRENT file's Message

151 152 153		RTS	; Block Address including Offsets (none ; Return to Exit routine	)
154	End		•	
155				
156		SAV	AWLOADERRD.OBJ	
157				

```
**********************
 1
 2
 3
   *
                    AWLOADERSD Merlin Source File
 4
                       - ShutDown (SD) File -
 5
              See Also Read (RD) File and SetUp (SU) File
 6
 7
                  (For use with FinderFileLoad Init)
 8
 9
                             by Hugh Hood
10
                          Version 1.3 (2024)
11
12
13
        based on the AppleWorks 3/UltraMacros 'FinderLaunch'
14
             by John L. Tegelaar and Ton W. van Santen
15
                      The Netherlands (1991)
16
17
18
19
               TR
                           ADR
                                        ; truncate bank address
20
21
               XC
                                        ; enable 65C02 code
22
               XC
                                        : enable 65C816 code
23
24
               ORG
                           $020F
                                        ; AppleWorks Load Location
25
               TYP
                           $06
                                        ; create binary file
26
27 UserID
               EQU
                           $203
                                        ; UserID assigned by Memory Manager
                                         Initial value is #$0000 (no message)
28
   MsqFlaq
               EQU
                           $205
29
   MsqCmnd
               EQU
                           $207
                                        ; Open ($0000) vs. Print ($0001)
30
                                         (both types are loaded to AppleWorks)
31 MsgOffset
               EQU
                           $209
                                        ; Offset into Message Block for filenames
32
                                          (initial value is #$0002 for File #1)
33 MsqBlock
               EQU
                           $20B
                                        ; Reading Address including Offsets
34
35 FullBuff
               EQU
                           $800
                                        ; fullpath work buffer
36 FileBuff
               EQU
                           $900
                                        ; filename work buffer
37
38 Handle
               EQU
                           $000
                                        ; in Bank $01 when UltraMacros is running
39
40
41 Begin
42
   Mode16
               CLC
                                         prep for native mode
43
                                        : now in native mode
               XCE
44
               REP
                           #$30
                                        ; 16-bit registers (X,Y,A)
45
46
47
                           UserID
                                        ; Load stored UserId and
               LDA
48
               PHA
                                           push it on stack (used later)
                                        ; Load high word of stored Handle and
49
               LDA
                           Handle+2
50
               PHA
                                           push it on stack (used later)
```

51 52 53 54 55 56 57 58 59 60 61 62	LDA PHA PEA PEA PHA PHA LDX JSL LDX JSL LDX	#\$1501 \$E10000 #\$1002 \$E10000 #\$0302	; Load low word of stored Handle and ; push it on stack (used later) ; Push Action #3 (Delete Message) on Stack ; Push Type/ID #1 (Filename) on Stack ; Push handle (any - # not important here) ; on Stack ; Delete from MessageCenter ; Tool Loader Routine ; Dispose Handle (Handle still on stack) ; Tool Loader Routine ; Memory Manager Shutdown ; (UserID still on stack)
63 64	JSL	\$E10000	; Tool Loader Routine
65	Mode8 SEC		; prep for emulation mode
66	XCE		; now in emulation mode
67	SEP	#\$30	; 8-bit registers (X,Y,A)
68			
69	Done RTS		; return to MACRO calling routine
70			
71	End		•
72			
73 74	SAV	AWLOADERSI	D.OBJ