```
*************************************************************
*                                                           *
*              WPSaveMore20 - An AppleWorks Init            *
*                                                           *
*       -   When saving Word Processor files as TEXT        *
*            files, this Init does (3) things:              *
*                                                           *
*             1. Fixes bug that caused TEXT files           *
*                saved with MS-DOS (CR/LF) line             *
*                enders to have a CR/LF pair appended       *
*                to the end of every 'screen' line,         *
*                not just those containing a <CR>.          *
*                                                           *
*             2. Supports saving TEXT files with            *
*                UNIX (LF) line enders.                     *
*                                                           *
*             3. Supports saving Merlin Source files        *
*                with high ASCII characters EXCEPT          *
*                <TAB> as $A0, and <SPACE> as $20.          *
*                                                           *
*                                                           *
*           Version 2.0 (for AppleWorks Version 5.1)        *
*                     (c) 2017 Hugh Hood                    *
*                                                           *
*     - Seg $29 (Save/Remove Files) is patch-in segment.    *
*                                                           *
*************************************************************

                TR              ADR             ; truncate bank address

                XC                              ; enable 65C02 code

* Equates *

TabChar         EQU             $A76            ; character to use for <TAB> in save routine
ORAChar         EQU             $A77            ; character to inclusive OR all characters
NewStr          EQU             $D85            ; used by WP to contain 'Line/Column' String
OldStr          EQU             $E05            ; used by WP to contain 'Line/Column' String
AWVersion       EQU             $1003           ; $33/51 = 5.1 / $28/40 = 4.0  /
                                                ;  $1E/30 = 3.0
MvLeftRtn       EQU             $1148           ; memory move / follow with TO/FROM/LENGTH
imSavePatch     EQU             $3006           ; Patch Manager save routine in SEG.IM
InitAdr         EQU             $4000           ; load address for Init files
PatchPoint1     EQU             $5FCA           ; first patch point in SEG $29
PatchPoint2     EQU             $5F47           ; second patch point in SEG $29
PatchPoint3     EQU             $5FC7           ; third patch point in SEG $29
PatchPoint4     EQU             $5E61           ; forth patch point in SEG $29
PatchPoint5     EQU             $5E50           ; fifth patch point in SEG $29
PatchPoint6     EQU             $5F22           ; sixth patch point in SEG $29
TabPatch        EQU             $5F69           ; use $A0 instead of $09 for Merlin TABs
ORAPatch        EQU             $5F78           ; ORA/set high bit on Merlin files
AddLineEnders   EQU             $5FFD           ; routine that adds line enders to
                                                ;  EVERY line
ModMMInDoc      EQU             $65FA           ; AWP file flag byte modified to reflect
                                                ;  state of 'save text files as text' flag
MMInDoc         EQU             $7C68           ; AWP file flag byte:
```

```
                                        ;  bit 1 (%00000001)/$01 = Mail Merge (AWP)
                                        ;  bit 2/5 (%00100100)/$24 = Merlin Source
                                        ;  bit 3/5 (%00101000)/$28 = Unix TEXT
                                        ;  bit 4/5 (%00110000)/$30 = MS-DOS TEXT
                                        ;  bit 5 (%00100000)/$20 = TEXT
Patch2Run       EQU         $6D00       ; final destination for new code
                                        ; NOTE: Seg $29 runs from $5200 - $65FC.
*******************************************************************************
* IMPORTANT NOTE: File Tags are normally loaded (one-by-one) from Desktop Memory
*                 into the unused space from $6600-$6DFF prior to being written out
*                 at the end of the file being saved. This space will support a
*                 maximum Tag size of $800/2048 bytes. This init will reduce the
*                 maximum possible Tag size to $700/1792 bytes. Fortunately, all (3)
*                 of the officially-registered AppleWorks file Tags are well below
*                 this limit:
*                  Word Processor Mail Merge filename Tag ($4D/M) - $13/19 bytes
*                  Word Processor Outliner styles Tag ($4F/O) - $25/37 bytes
*                  TimeOut Graph selections Tag ($04) - $1FC/252 bytes
*                 If you were to write 'custom' AppleWorks modules, inits, or
*                 macros, and were to use file Tags, you need to keep the length
*                 of those Tags to a maximum of $700/1792 bytes.
*******************************************************************************

PatchAdr        EQU         $BB00       ; load address for patch code
                                        ;  (NOTE: uses ProDOS I/O buffer -
                                        ;          1K max length -
                                        ;           $BB00 - $BEFF)
                                        ;
                ORG         InitAdr     ; ($4000)
                TYP         $06         ; create binary file

*********************************
*          Init Header          *
*********************************
START
                JMP         IStart      ; skip over header

**-------------------------------------------------------------

                ASC         'mb'        ; Init ID Bytes (AW 5.1)
                DB          $14         ; Init Version - programmer assigned
                                        ;  e.g. - $0A/1.0 $0B/1.1 $19/2.5
                STR         'WPSaveMore20' ; Init Screen Name
                HEX         0000        ; Header End Bytes

**-------------------------------------------------------------

IStart

                LDA         AWVersion   ; AppleWorks version #
                CMP         #$33        ; Is it Version 5.1?
                BNE         Done        ; disregard - wrong version

PatchH29        JSR         imSavePatch ; call patch manager
                DW          Code1       ; beginning of patch1 code ($40xx)
                DW          Patch1End-PatchAdr+Patch2End-Patch2Run
```

```
; length of patch code

                DW              $0029           ; SEG number to patch
                                                ;  ($29 = Save/Remove Files SEG)

Done            RTS                             ; back to Init Manager

**--------------------------------------------------------------

Code1           EQU             *               ; (will be $40xx)

                ORG             PatchAdr        ; (Patching Code is moved and run
                                                ;   @ $BB00 by Init Manager)

HookBytes       HEX             0000            ; first bytes for $29 Patch

                LDA             #$4C            ; JMP instruction
                STA             PatchPoint1     ; $5FCA in SEG $29
                LDA             #Patch2Run      ; low byte of new code
                STA             PatchPoint1+1
                LDA             #>Patch2Run     ; high byte of new code
                STA             PatchPoint1+2

                LDA             #$1C            ; MS-DOS/Unix or Merlin? (%00011100)
                STA             PatchPoint2     ; was just MS-DOS (%00010000)
                STA             PatchPoint3     ; was just MS-DOS (000010000)

                LDA             #$C3            ; make low TXT/DOS/Unix/Merlin bits
                                                ;  (%11000011)
                STA             PatchPoint4     ; was just TXT/MS-DOS (%11001111)

                LDA             #$3C            ; check for TXT/DOS/Unix/Merlin bits
                                                ;  (%00111100)
                STA             PatchPoint5     ; was just TXT/MS-DOS (%00110000 - $30)

                LDA             #$4C            ; JMP instruction
                STA             PatchPoint6     ; $5F22 in SEG $29
                LDA             #CROnly         ; low byte of new code
                STA             PatchPoint6+1
                LDA             #>CROnly        ; high byte of new code
                STA             PatchPoint6+2

                LDA             #$4C            ; JMP instruction
                STA             TabPatch        ; $5F69 in SEG $29
                LDA             #TabCode        ; low byte of new code
                STA             TabPatch+1
                LDA             #>TabCode       ; high byte of new code
                STA             TabPatch+2

                LDA             #$4C            ; JMP instruction
                STA             ORAPatch        ; $5F78 in SEG $29
                LDA             #ORACode        ; low byte of new code
                STA             ORAPatch+1
                LDA             #>ORACode       ; high byte of new code
                STA             ORAPatch+2
```

```
168
169 MoveCall       JSR             MvLeftRtn       ; move new code to run location
170                 DA              #Patch2Run      ; ($6D00)
171                 DA              #MoveStart      ;
172                 DA              Patch2End-Patch2Run
173
174                 RTS                             ; patch hook-in done
175
176 Patch1End      EQU             *
177
178
179 **--------------------------------------------------------------
180
181 MoveStart      EQU             *               ; (will be $BBxx)
182
183                 ORG             Patch2Run       ; ($6D00)
184
185
186 * MS-DOS/UNIX lines that include a <CR>
187
188 PatchStart     LDA             NewStr+1
189                 BPL             GoBack1         ; no <CR> on this line
190                                                 ;  (do NOT add line enders)
191                 LDA             ModMMInDoc
192                 AND             #$08            ; Unix Bit Set (Bit 3)
193                 BEQ             :A
194                 LDA             #$0A
195                 STA             AddLineEnders+1
196                 BRA             :B
197
198 :A             LDA             ModMMInDoc
199                 AND             #$04            ; Merlin Bit Set (Bit 2)
200                 BEQ             :B              ; Not Unix / Not Merlin
201                 LDA             #$8D            ; Merlin Line Ender
202                 STA             AddLineEnders+1
203
204 :B             JSR             AddLineEnders
205
206                 LDA             #$0D
207                 STA             AddLineEnders+1
208
209 GoBack1        JMP             PatchPoint1+3   ; go back to original code ($5FCD)
210
211 **--------------------------------------------------------------
212
213 * MS-DOS/UNIX/Merlin lines containing ONLY a <CR>
214
215 CROnly         LDA             ModMMInDoc
216                 AND             #$08            ; Unix Bit Set (Bit 3)
217                 BEQ             :C              ; Not Unix
218                 LDA             #$0A            ; Unix Line Ender
219                 STA             AddLineEnders+1
220                 BRA             :D
221
222 :C             LDA             ModMMInDoc
223                 AND             #$04            ; Merlin Bit Set (Bit 2)
```

```
                BEQ             :D              ; Not Unix / Not Merlin
                LDA             #$8D            ; Merlin Line Ender
                STA             AddLineEnders+1

:D              JSR             AddLineEnders   ; do routine to add proper line enders
                                                ;  to <CR> ONLY lines

                LDA             #$0D            ; original value
                STA             AddLineEnders+1              ; restore original value

GoBack2         JMP             PatchPoint6+3   ; go back to original code ($5F25)


**--------------------------------------------------------------

TabCode         LDA             ModMMInDoc
                AND             #$04            ; Merlin Bit Set (Bit 2)
                BEQ             :E              ; Not Merlin
                LDA             #$A0            ; high bit space for Merlin <TAB>
                BRA             :F              ; get back to character parser

:E              LDA             TabChar         ; 'normal' <TAB> substitution character

:F              JMP             TabPatch+3


ORACode         PHA                             ; save accumlator (character)
                LDA             ModMMInDoc
                AND             #$04            ; Merlin Bit Set (Bit 2)
                BEQ             :G              ; Not Merlin
                PLA                             ; retrieve character
                ORA             #$80            ; set high bit for Merlin character
                BRA             :H              ; get back to character parser

:G              PLA                             ; retrieve character
                ORA             ORAChar         ; normally '$00' / '$80 sets high bit

:H              JMP             ORAPatch+3

**--------------------------------------------------------------

Patch2End       EQU             *               ;
                SAV             I.WPSAVEMORE20
                LST             OFF

                END
```