

```

1 *****
2 *
3 *      WPUntixLoad20 - An AppleWorks Init
4 *
5 *      - When loading Text Files with UNIX-style
6 *      line ends (LF only), this Init
7 *      properly translates the LF characters
8 *      into line ends used by the AppleWorks
9 *      word processor. It also sets the 'UNIX'
10 *      flag so that subsequent saves of the
11 *      will use the LF character as the line
12 *      ender (provided the 'WPSaveMore' init
13 *      is also installed and active).
14 *
15 *
16 *      Version 2.0 (for AppleWorks Version 5.1)
17 *      (c) 2017 Hugh Hood
18 *
19 *      - Seg $28 (Load Files) is patch-in segment.
20 *
21 *****
22
23          TR          ADR          ; truncate bank address
24
25          XC          ; enable 65C02 code
26
27 * Equates *
28
29 AWWVersion      EQU          $1003          ; $33/51 = 5.1 / $28/40 = 4.0 /
30                                     ; $1E/30 = 3.0
31 MvLeftRtn      EQU          $1148          ; memory move / follow with TO/FROM/LENGTH
32 StrMvRtn       EQU          $116C          ; string move / follow with TO/FROM
33 imSavePatch    EQU          $3006          ; Patch Manager save routine in SEG.IM
34 InitAdr        EQU          $4000          ; load address for Init files
35 PatchPoint1    EQU          $6495          ; first patch point in SEG $28
36 CRHandler      EQU          $6533          ; <CR> handler/storage routine on load
37 MMInDoc        EQU          $7C68          ; AWP file flag byte:
38                                     ; bit 0 (%00000001)/$01 = Mail Merge (AWP)
39                                     ; bit 2/5 (%00100100)/$24 = Merlin Source
40                                     ; bit 3/5 (%00101000)/$28 = Unix TEXT
41                                     ; bit 4/5 (%00110000)/$30 = MS-DOS TEXT
42                                     ; bit 5 (%00100000)/$20 = TEXT
43 Patch2Run      EQU          $6D90          ; final destination for new code
44                                     ; NOTE: Seg $28 runs from $5200 - $65F6,
45                                     ; and uses some space following
46                                     ; $6600.
47 *****
48 * IMPORTANT NOTE: File Tags are normally loaded (one-by-one) from the file into the
49 *      unused space from $6600-$6DFF prior to being saved to Desktop
50 *      Memory with a pointer stored in the Tag Table. This space will
51 *      support a maximum Tag size of $800/2048 bytes. This init will
52 *      reduce the maximum possible Tag size to $700/1792 bytes.
53 *      Fortunately, all (3) of the officially-registered AppleWorks file
54 *      Tags are well below this limit:
55 *      Word Processor Mail Merge filename Tag ($4D/M) - $13/19 bytes
56 *      Word Processor Outliner styles Tag ($4F/O) - $25/37 bytes

```

```

57 *           TimeOut Graph selections Tag ($04) - $1FC/252 bytes
58 *           If you were to write 'custom' AppleWorks modules, inits, or
59 *           macros, and were to use file Tags, you need to keep the length
60 *           of those Tags to a maximum of $700/1792 bytes.
61 *****
62
63 PatchAdr      EQU          $BB00          ; load address for patch code
64                                     ; (NOTE: uses ProDOS I/O buffer -
65                                     ;           1K max length -
66                                     ;           $BB00 - $BEFF)
67                                     ;
68                                     ;
69                                     ; ($4000)
69                                     ; create binary file
70
71 *****
72 *           Init Header          *
73 *****
74 START
75             JMP          IStart          ; skip over header
76
77 **-----
78
79             ASC          'mb'           ; Init ID Bytes (AW 5.1)
80             DB           $14           ; Init Version - programmer assigned
81                                     ; e.g. - $0A/1.0 $0B/1.1 $19/2.5
82             STR          'WPUnixLoad20' ; Init Screen Name
83             HEX          0000         ; Header End Bytes
84
85 **-----
86
87 IStart
88
89             LDA          AWVersion      ; AppleWorks version #
90             CMP          #$33          ; Is it Version 5.1?
91             BNE          Done          ; disregard - wrong version
92
93 PatchH28      JSR          imSavePatch  ; call patch manager
94             DW           Code1         ; beginning of patch1 code ($40xx)
95             DW           Patch1End-PatchAdr+Patch2End-Patch2Run
96             ; length of patch code
97
98             DW           $0028         ; SEG number to patch
99                                     ; ($28 = Load Files SEG)
100
101 Done          RTS                    ; back to Init Manager
102
103 **-----
104
105 Code1         EQU          *           ; (will be $40xx)
106
107             ORG          PatchAdr      ; (Patching Code is moved and run
108                                     ; @ $BB00 by Init Manager)
109
110 HookBytes     HEX          0000         ; first bytes for $28 Patch
111

```

```

112          LDA          #$4C          ; JMP instruction
113          STA          PatchPoint1  ; $6495 in SEG $28
114          LDA          #Patch2Run   ; low byte of new code
115          STA          PatchPoint1+1
116          LDA          #>Patch2Run  ; high byte of new code
117          STA          PatchPoint1+2
118
119
120 MoveCall   JSR          MvLeftRtn   ; move new code to run location
121          DA          Patch2Run     ; ($6D90)
122          DA          MoveStart     ;
123          DA          Patch2End-Patch2Run
124
125          RTS                    ; patch hook-in done
126
127 Patch1End  EQU          *
128
129
130 **-----
131
132 MoveStart  EQU          *          ; (will be $BBxx)
133
134          ORG          Patch2Run    ; ($6D90)
135
136
137 * Check Text file for <LF> ONLY line endings
138
139 PatchStart BNE          SetUnix     ; has <LF> but no preceding <CR>
140
141 SetDOS     LDA          #%00110000  ; #$30 (keep TXT/#$20 Bit)
142          JMP          PatchPoint1+4 ; go back to original code ($6499)
143
144 SetUnix    LDA          #%00101000  ; #$28 (keep TXT/#$20 Bit)
145          STA          MMInDoc
146          JMP          CRHandler    ; register <CR> and read next line
147
148
149 **-----
150
151
152 Patch2End  EQU          *          ;
153          SAV          I.WPUNIXLOAD20
154          LST          OFF
155
156          END
157
158

```