

```

1 *****
2 *
3 *          SSSStatus - An AppleWorks Init          *
4 *
5 *          - Adds Real Time File Status Indicator to *
6 *          Spreadsheet REVIEW/ADD/CHANGE Window - *
7 *
8 *          - 'New' or 'Changed' File Indicated by Mousetext *
9 *          Diamond Character in place of colon after *
10 *          'File' in top line of Display.          *
11 *
12 *          Version 2.3 (for AppleWorks Version 5.1) *
13 *          (c) 2015 Hugh Hood                      *
14 *
15 *          - Seg $1E is patch-in segment, but NEW code requires *
16 *          space at end of Seg $1D (Main), because placing *
17 *          code at either $BB00 (default) or at most other *
18 *          locations results in it being overwritten by *
19 *          newly-loaded Segments. By moving the new code *
20 *          to the Main SS Segment patch, we ensure that it *
21 *          is always present whenever SpreadSheet is the *
22 *          active application, since Seg $1D must be loaded. *
23 *
24 *****
25
26          TR          ADR          ; truncate bank address
27
28          XC          ; enable 65C02 code
29
30 * Equates *
31
32 MTDiamond      EQU          $DB          ; mousetext 'diamond' character
33 DTFStatus      EQU          $C6C        ; current file status flag byte:
34                ; ($01/0001) or $03/0011) - new
35                ; ($02/0010) or $06/0110) - changed
36                ; ($04/0100) - saved
37                ; ($00/0000) - unchanged
38 LJCellID       EQU          $802C       ; used by SS to contain CellID (e.g. 'A1')
39 AWVersion      EQU          $1003       ; $33/51 = 5.1 / $28/40 = 4.0 /
40                ; $1E/30 = 3.0
41 MvLeftRtn     EQU          $1148       ; memory move / follow with TO/FROM/LENGTH
42 StrWrRtn      EQU          $116F       ; writes a string at a fixed location
43                ; on the screen
44                ; (follow JSR with col/row/string address)
45 imSavePatch    EQU          $3006       ; Patch Manager save routine in SEG.IM
46 InitAdr       EQU          $4000       ; load address for Init files
47 PatchPoint    EQU          $3D5E       ; patch point in SEG $1E
48 Patch2Run     EQU          $7ED0       ; final destination for new code
49                ; NOTE: Seg $1E runs from $3900 - $4EFE,
50                ; but seems to use memory up through
51                ; $51FF - Other SpreadSheet Segs
52                ; load at $5300, $7400 and $2100
53                ; - not even $5200 is free for long -
54                ; - Seg $1D runs from $7400-$7EC3
55                ; - SS work area starts $7F00
56 PatchAdr      EQU          $BB00       ; load address for patch code

```

```

57                                     ; (NOTE: uses ProDOS I/O buffer -
58                                     ;           1K max length -
59                                     ;           $BB00 - $BEFF)
60
61
62          ORG          InitAdr          ; ($4000)
63          TYP          $06              ; create binary file
64
65 *****
66 *           Init Header           *
67 *****
68 START
69          JMP          IStart           ; skip over header
70
71 **-----
72
73          ASC          'mb'            ; Init ID Bytes (AW 5.1)
74          DB           $17             ; Init Version - programmer assigned
75                                     ; e.g. - $0A/1.0 $0B/1.1 $17/2.3
76          STR          'SSStatus'      ; Init Screen Name
77          HEX          0000           ; Header End Bytes
78
79 **-----
80
81 IStart
82
83          LDA          AWVersion        ; AppleWorks version #
84          CMP          #$33            ; Is it Version 5.1?
85          BNE          Done            ; disregard - wrong version
86
87 PatchH1E   JSR          imSavePatch    ; call patch manager
88          DW          Code1            ; beginning of patch1 code ($40xx)
89          DW          Patch1End-PatchAdr
90          ; length of patch code
91          DW          $001E           ; SEG number to patch
92                                     ; ($1E = ASP Edit SEG)
93
94 PatchH1D   JSR          imSavePatch    ; call patch manager
95          DW          Code1+Patch1End-PatchAdr
96          ; beginning of patch2 code ($40xx)
97          DW          MoveStart-PatchAdr+Patch2End-Patch2Run
98          ; length of patch code
99          DW          $001D           ; SEG number to patch
100                                     ; ($1D = ASP Main SEG)
101
102 Done       RTS
103
104 **-----
105
106 Code1      EQU          *             ; (will be $40xx)
107
108          ORG          PatchAdr        ; (Patching Code is moved and run
109                                     ; @ $BB00 by Init Manager)
110
111 HookBytes  HEX          0000         ; first bytes for $1E Patch

```

```

110          LDA          #$4C          ; JMP instruction
111          STA          PatchPoint    ; $3D5E in SEG $1E
112          LDA          #Patch2Run    ; low byte of new code
113          STA          PatchPoint+1
114          LDA          #>Patch2Run   ; high byte of new code
115          STA          PatchPoint+2
116
117          RTS
118
119 Patch1End  EQU          *
120
121 **-----
122
123          ORG          PatchAdr       ; (Patching Code is moved and run
124                                     ; @ $BB00 by Init Manager)
125
126 MoveCall  HEX          0000         ; first bytes for $1D Patch
127          JSR          MyLeftRtn     ; move new code to run location
128          DA          #Patch2Run     ; ($7ED0)
129          DA          #MoveStart     ;
130          DA          Patch2End-Patch2Run
131
132          RTS                      ; patch hook-in done
133
134 **-----
135
136 MoveStart EQU          *           ; (will be $BBxx)
137
138          ORG          Patch2Run     ; ($7ED0)
139
140 PatchStart LDA         #' :'       ; unchanged 'File' suffix
141          STA         StatusStr+1    ;
142          LDA         DTFStatus      ; current file status flag byte
143          AND         #%00000011    ; test if either Bit 0 or Bit 1 is set
144                                     ; to check for unchanged or saved
145          BEQ         :A             ; if both bits are clear, branch around
146          LDA         #MTDiamond     ; mousetext 'diamond' suffix
147          STA         StatusStr+1    ;
148 :A        JSR         StrWrRtn      ;
149          DB         $04             ; column $04/04
150          DB         $00             ; row $00/00
151          DA         StatusStr       ; either colon or MTDiamond character
152          JSR         StrWrRtn      ; originally from $3D5E-3D60
153          DB         $00             ; column $00/00 {was $3D61}
154          DB         $15             ; row $15/21 {was $3D62}
155          DA         LJCellID        ; used by SS to contain CellID (e.g. 'A1')
156                                     ; {was $3D63-3D64}
157
158          JMP         PatchPoint+7   ; go back to original code
159
160 **-----
161
162 StatusStr Str         ' :'         ; default 'File' suffix
163
164 **-----
165

```

```
166 Patch2End EQU * ;
167 SAV I.SSSTATUS
168 LST OFF
169
170 END
171
```