

```

*****
*
*           DBStatus - An AppleWorks Init
*
*   - Adds Real Time File Status Indicator to
*     Data Base REVIEW/ADD/CHANGE Window -
*
*   - 'New' or 'Changed' File Indicated by Mousetext
*     Diamond Character in place of colon after
*       'File' in top line of Display.
*
*           Version 2.3 (for AppleWorks Version 5.1)
*           (c) 2015 Hugh Hood
*
*   - Seg $02 is patch-in segment, but NEW code requires
*     space at end of Seg $01 (Main), because placing
*     code at either $BB00 (default) or at most other
*     locations results in it being overwritten by
*     newly-loaded Segments. By moving the new code
*     to the Main DB Segment patch, we ensure that it
*     is always present whenever Data Base is the
*     active application, since Seg $01 must be loaded.
*
*****

```

```

TR      ADR      ; truncate bank address

```

```

XC      ; enable 65C02 code

```

```

* Equates *

```

```

CurRec      EQU      $B4      ; current data base record # (word)
MTDiamond    EQU      $DB      ; mousetext 'diamond' character
DTFStatus    EQU      $C6C     ; current file status flag byte:
                                ; ($01/0001) or $03/0011) - new
                                ; ($02/0010) or $06/0110) - changed
                                ; ($04/0100) - saved
                                ; ($00/0000) - unchanged
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
StrWrRtn     EQU      $116F    ; writes a string at a fixed location
                                ; on the screen
                                ; (follow JSR with col/row/string address)
imSavePatch  EQU      $3006    ; Patch Manager save routine in SEG.IM
InitAdr      EQU      $4000    ; load address for Init files
PatchPoint   EQU      $251A    ; patch point in SEG $02
                                ; (routine to print current record
                                ; # on screen)
Patch2Run    EQU      $91D0    ; final destination for new code
                                ; NOTE: Seg $02 runs from $2000 - $5105.
                                ; Other Data Base Segs load at
                                ; $3900, $3D00, $4000, $4100,
                                ; $4B00, $5200 and $5A00
                                ; - Seg $01 runs from $7800-$91B5
                                ; - DB work area starts $9200
PatchAdr     EQU      $BB00    ; load address for patch code
                                ; (NOTE: uses ProDOS I/O buffer -
                                ; 1K max length -
                                ; $BB00 - $BEFF)

```

```

Word2Str      EQU      $D044      ; convert Hex word to Decimal string

                                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       $17         ; Init Version - programmer assigned
                                                ; e.g. - $0A/1.0 $0B/1.1 $17/2.3
                                STR      'DBStatus'   ; 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

PatchH02      JSR      imSavePatch ; call patch manager
                                DW       Code1        ; beginning of patch1 code ($40xx)
                                DW       Patch1End-PatchAdr
                                                ; length of patch code
                                DW       $0002        ; SEG number to patch
                                                ; ($02 = ADB Edit SEG)

PatchH01      JSR      imSavePatch ; call patch manager
                                DW       Code1+Patch1End-PatchAdr
                                                ; beginning of patch2 code ($40xx)
                                DW       MoveStart-PatchAdr+Patch2End-Patch2Run
                                                ; length of patch code
                                DW       $0001        ; SEG number to patch
                                                ; ($01 = ADB Main 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 $02 Patch
                                LDA      #$4C         ; JMP instruction
                                STA      PatchPoint   ; $251A in SEG $02
                                LDA      #Patch2Run   ; low byte of new code
                                STA      PatchPoint+1

```

```

        LDA    #>Patch2Run
        ; high byte of new code
        STA    PatchPoint+2

        RTS

Patch1End    EQU    *

**-----

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

MoveCall      HEX    0000          ; first bytes for $01 Patch
              JSR    MvLeftRtn    ; move new code to run location
              DA     #Patch2Run    ; ($91D0)
              DA     #MoveStart    ;
              DA     Patch2End-Patch2Run

              RTS                  ; patch hook-in done

**-----

MoveStart     EQU    *            ; (will be $BBxx)

                ORG    Patch2Run    ; ($91D0)

PatchStart    LDA    #':'          ; unchanged 'File' suffix
              STA    StatusStr+1    ;
              LDA    DTFStatus      ; current file status flag byte
              AND    #%00000011     ; test if either Bit 0 or Bit 1 is set
                                ; to check for unchanged or saved
              BEQ    :A             ; if both bits are clear, branch around
              LDA    #MTDiamond     ; mousetext 'diamond' suffix
              STA    StatusStr+1    ;
:A            JSR    StrWrRtn        ;
              DB     $04            ; column $04/04
              DB     $00            ; row $00/00
              DA     StatusStr      ; either colon or MTDiamond character
              JSR    Word2Str        ; originally from $251A-251C
              DA     CurRec          ; originally from $251D-241E

              JMP    PatchPoint+5    ; go back to original code at $251F

**-----

StatusStr     Str    ':'           ; default 'File' suffix

**-----

Patch2End     EQU    *            ;
              SAV    I.DBSTATUS
              LST    OFF

              END

```