

```

*****
*
*           MaxDesk - An AppleWorks Init
*
*   - When quitting AppleWorks running under
*     ProDOS 8 on the Apple IIGS, the
*     MaxDesk Init disposes of the memory
*     allocated by the Apple IIGS Memory
*     Manager to the AppleWorks Desktop
*     so that restarting AppleWorks in a
*     single session does not result in
*     a diminished Desktop size.
*     (approximately 230kB reduction)
*
*           - BACKGROUND -
*
*   Under ProDOS 8 (no tool patches) the
*   DisposeAll tool does not seem reliably to
*   work if the assigned Application Memory has
*   an Aux ID of 1 (the second digit in $11xx),
*   unless an Aux ID of 0 (wildcard) is used in
*   the DisposeAll call. This Init patches
*   AppleWorks to use an Aux ID of 0 when quitting,
*   thus correcting the problem.
*
*   - NOTE: This init is not required when running
*     AppleWorks under GS/OS, but its
*     presence in the AW.INITS folder
*     causes no harm.
*
*           Version 1.0 (for AppleWorks Version 5.1)
*           (c) 2015 Hugh Hood
*
*   - Seg $29 is patch-in segment, but only the exiting
*     code is modified. No NEW code is required.
*
*****

```

```

TR      ADR      ; truncate bank address

```

```

XC      ; enable 65C02 code

```

```

XC      ; enable 65816 code

```

```

* Equates *

```

```

MemoryID    EQU    $0FDE    ; Memory ID# stored here by AppleWorks
Seg00Type   EQU    $FE0     ; memory manager type
                        ; $41 = SEG.AM / $52 = SEG.RM (IIGS) /
                        ; $58 = SEG.XM / $4D = Deja IIX (Mac)
AWVersion   EQU    $1003    ; $33/51 = 5.1 / $28/40 = 4.0 /
                        ; $1E/30 = 3.0
DisposeAll  EQU    $1102    ; DisposeAll Tool for Memory Manager
MvLeftRtn   EQU    $1148    ; memory move / follow with TO/FROM/LENGTH
DeleteID    EQU    $2103    ; DeleteID Tool for Memory Manager
imSavePatch EQU    $3006    ; Patch Manager save routine in SEG.IM
InitAdr     EQU    $4000    ; load address for Init files

```

```

PatchPoint    EQU    $65D0        ; patch point in SEG $29
                                   ; (routine to Dispose of memory allocated
                                   ;   by Apple IIGS Memory Manager)
PatchAdr       EQU    $BB00        ; load address for patch code
                                   ; (NOTE: uses ProDOS I/O buffer -
                                   ;       1K max length -
                                   ;       $BB00 - $BEFF)
ToolLocate     EQU    $E10000      ; GS Tool Locator Call

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

                LDA     Seg00Type    ; Memory Manager Type
                CMP     #$52         ; Is it Apple IIGS? (SEG.RM)
                BNE     Done         ; disregard - IIGS only

PatchH29       JSR     imSavePatch  ; call patch manager
                DW      Code1        ; beginning of patch1 code ($40xx)
                DW      MoveStart-PatchAdr+Patch1End-PatchPoint
                                   ; length of patch code
                DW      $0029        ; SEG number to patch
                                   ; ($29 = Organizer Save/Remove 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 Patch
                JSR     MvLeftRtn    ; move new code to run location
                DA      #PatchPoint  ; ($65D0)
                DA      #MoveStart   ;
                DA      Patch1End-PatchPoint

```

```

                RTS                ; patch done

**-----

MoveStart      EQU      *          ; (will be $BBxx)

                ORG      PatchPoint ; ($65D0)

                MX       %00        ; 16-bit ACC and Registers

PatchStart     AND      #$F0FF     ; force Aux ID# to 0
                STA      MemoryID   ; store modified Aux ID#
                PHA                     ; push the ID# on the stack
                NOP                     ; kills an unused patched byte
                LDX      #DisposeAll ; ready DisposeAll Tool
                JSL      ToolLocate  ; Tool Locator call
                LDA      MemoryID   ; modified by AND
                PHA                     ; push the ID# on the stack
                LDX      #DeleteID  ; ready DeleteID Tool

**-----

                MX       %11        ; 8-bit ACC and Registers

Patch1End      EQU      *          ;
                SAV      I.MAXDESK
                LST      OFF

                END

```