```
*************************************************************
*                                                           *
*                      TimeOut EditBASIC                    *
*                     for AppleWorks 5.1                    *
*                                                           *
*          - Load, View and Edit AppleSoft BASIC            *
*             programs from within AppleWorks -             *
*                                                           *
*                        Version 5.4                        *
*                     February 25, 2019                     *
*                                                           *
*                       by Hugh Hood                        *
*                                                           *
*                                                           *
*                                                           *
*************************************************************

               TR             ADR            ; truncate bank address

               XC                            ; enable 65C02 code


* Internal AppleWorks Equates
ErrFlag        EQU            $03            ; ProDOS error occur?
FoundEsc       EQU            $88            ; non-zero if <ESC> pressed
DTCurOpen      EQU            $C54           ; current file # (if any)
DTCtOnDesk     EQU            $C55           ; number of files on current Desktop
NewStr         EQU            $D85           ; 128 bytes (used by many routines)
OrgInMem       EQU            $E86           ; $00 = in file / $01 = in menu /
                                             ;   $40 = in file list
StrWork5       EQU            $FD7           ; 5-byte pString used by newWord2Str


* AppleWorks Host Equates
AWVersion      EQU            $1003          ; $33/51 = 5.1 / $28/40 = 4.0  /
                                             ;   $1E/30 = 3.0
PrevSeg        EQU            $1021          ; SEG in use (last loaded by CallSeg)
CallSeg        EQU            $10FA          ; AW segment loading routine
                                             ;  - LDA with segment # before calling
                                             ;     ($01/01 - $2F/47)
                                             ;  - see segment chart
                                             ;  - add $80/128 to load only (not JMP)
ClearDA        EQU            $1100          ; clears display area on screen
DoGoToXY       EQU            $1118          ; gotoXY (cursor on screen)
MvRightRtn     EQU            $114B          ; move in memory routine (positive/right)
                                             ; - followed by destination/source/length
PressAny       EQU            $1151          ; 'Press space to continue'
PushStack      EQU            $1157          ; add a level to Escape road map
                                             ; - followed by address of text pString
StrMvRtn       EQU            $116C          ; copy a pString
                                             ; - followed by destination/source
WritePRtn      EQU            $1187          ; propagates a character (repeats)
                                             ;  (X) times of (Y) character
DrawBox        EQU            $11BA          ; draws rectangle mousetext box on screen
                                             ; - followed by column/row/width/height

```

```
57
58  * AppleWorks SEG $27 (Organizer Main) Equates
59  DrawFileCard    EQU             $211A           ; draw AppleWorks FileCard
60                                                  ; - precede with level # in (A)
61  SlotClrCurr     EQU             $2132           ; routine to clear (zero out) contents of
62                                                  ;  "current" file slot from $C56-$C75
63                                                  ; (DTFName/Path/Type/Status/Picked/Size)
64  SlotFrCurr      EQU             $2135           ; routine to save the data about the
65                                                  ;  "current" file back into its
66                                                  ;  proper slot (1-12)
67  Slot2Curr       EQU             $2138           ; routine to load & open one of the
68                                                  ;  desktop files (1-12) and make it the
69                                                  ;  "current" file
70  ParmTblPtr      EQU             $3B28           ; Address of ParmTable Pointer
71  RealTypeNew     EQU             $431E           ; 'Type a new name' routine
72  PromptNew       EQU             $432F           ; Address of TypeNew Prompt String
73  EnterFile       EQU             $4C35           ; Enter file just loaded
74  DTCountRtn      EQU             $4D1A           ; Ensures no more than 12/36 Files
75  GoMainEntry     EQU             $4D3B           ; Goto Main Menu and Wait for Input
76  NewFrOther      EQU             $4D99           ; Make WP/Text from 'Other'
77  TypeNew         EQU             $4DBE           ; call 'Type a new name' routine
78  TryAgain        EQU             $4DF3           ; Back to Add Files Menu after <ESC> Load
79
80  * AppleWorks SEG $28 (File Loader) Equates
81  RemFilesRtn     EQU             $5D25           ; Adds file to Remembered Files
82  TextLoader      EQU             $6366           ; Text Loader in SEG $28
83  SetFileStatus   EQU             $65A1           ; set status of loaded 'Text' file
84
85  * AppleWorks SEG $15 (Word Processor Main) Equates
86  WPPutLine       EQU             $6E1A           ; stores WpWa/$7B00 work register line
87                                                  ;  in desktop memory
88
89
90  * AppleWorks Subhost Equates
91  WriteText       EQU             $AB37           ; routine to write multiple strings
92                                                  ;  - followed by address of strings 'block'
93                                                  ;  - block contains:   (i) pString for 1
94                                                  ;                     (ii) Column for 1
95                                                  ;                    (iii) Row for 1
96                                                  ;                     (iv) repeat i/ii/iii
97                                                  ;                      (v) $00 = endbyte
98  newWord2Str     EQU             $AB61           ; convert HEX word to decimal
99                                                  ;  (result in StrWork5) [5 character max]
100
101
102 * AppleWorks TimeOut Engine Equates
103 AccAdr          EQU             $2100           ; TimeOut Application Loads Here to run
104 TOOrg          EQU             $B80A           ; TimeOut Organizer Loaded?
105                                                 ;  $00 = No / $01 = Yes
106
107 * TimeOut Edit BASIC Equates
108 MoveAdrs        EQU             $7E00           ; Relocate application here
109                                                 ;  (above SEG $27, SEG $28 and SEG $15)
110
111
112
```

```
113 ************************************************************
114
115               ORG             AccAdr          ; $2100
116
117 Start
118               JMP             MainSeg         ; skip over TimeOut header
119
120
121 *============================================================
122 * Timeout Initialization Header
123 *============================================================
124
125 IdBytes         HEX             0B02E12644      ; TimeOut unique signature bytes
126 MVersion        HEX             10              ; Minimum TimeOut version required (1.0)
127 MemRes          DB              0               ; Memory resident (T/F)
128 LockMR          DB              0               ; Lock MemRes flag (T/F)
129
130 * If your accessory uses any hires graphics, set the following bits in the
131 *  'Graphics' byte for correct memory allocation
132 Graphics        DB              0               ; Bit 0:Hi-Res page 1
133                                                 ;     1:Hi-Res page 2
134                                                 ;     2:Dbl hi-res page 1
135                                                 ;     3:Dbl hi-res page 2
136                                                 ;     4:Super (GS) hi-res
137
138 * Maximum 16 character application name
139 Name            STR             'Edit BASIC 5.4'
140               DS              Name-*+17       ; Pad so exactly 17 bytes for name
141
142 * The Timeout Engine puts the pathname of the accessory here
143 * so that the accessory can reload itself or its segments.
144 * NOTE: Move if area will be overwritten
145 PathName        DS              65
146
147 Preserve        DS              2               ; These two bytes contain the same value
148                                                 ;  that was left here the previous time this
149                                                 ;  accessory was called.
150 ConfigSeg       DB              2               ; Configuration segment number if applicable
151 ID              DB              $FF             ; ID's were assigned by Beagle Bros
152                                                 ;  (if not assigned = $FF)
153 Version         DB              54              ;  Version 5.4
154               DS              2
155 Reserved        DS              9               ; Must be 0's
156 NumSegs         DB              2               ; Number of segments >= 1
157 Offset1         ADR             $000000         ; Offset of Segment 1 from Start of File
158 Ptr1            DA              $0000           ; Filled in by Timeout if memory resident
159 Offset2         ADR             Config-Start    ; Offset of Segment 2 from Start of File
160 Ptr2            DA              $0000           ; Filled in by TimeOut if memory resident
161 EOF             DA              CodeEnd-Start   ; Length of complete TimeOut
162               DA              $0000           ; Header END bytes
163
164 *------------------------------------------------------------
165 * Main Segment
166 *------------------------------------------------------------
167
168 MainSeg
```

```
                JSR             ClearDA         ; clears display area on screen
                
* Version Check for AppleWorks 5.1
                LDA             AWVersion       ; version # of this AppleWorks
                CMP             #$33            ; 5.1
                BCC             NoBox           ; not version 5.1 - skip box
                
* Passed Version Check - Can Draw Box
                JSR             DrawBox         ; AppleWorks 5.1 routine
                DB              #$0A            ; column (10)
                DB              #$03            ; row (3)
                DB              #$3A            ; width (58)
                DB              #$0B            ; height (11)
                
* Write Text for Splash Screen
NoBox           LDX             #$14            ; column (20)
                LDY             #$06            ; row (6)
                JSR             DoGoToXY        ; gotoXY (cursor on screen)
                LDX             #$28            ; width (40)
                LDY             #$CC            ; MouseText Top Bar character
                JSR             WritePRtn       ; draw MT Top Bar on screen at cursor
                
                JSR             WriteText       ; routine to write multiple strings
                DA              SplashScreenA   ; address of strings block
                
* Version Check for AppleWorks 5.1
                LDA             AWVersion       ; version # of this AppleWorks
                CMP             #$33            ; 5.1
                BCC             NoGo            ; not version 5.1 - message and Quit
                
* Passed Version Check - Finish Splash Screen & Continue with Application
                JSR             WriteText       ; routine to write multiple strings
                DA              SplashScreenB   ; address of strings block
                
                JSR             PressAny        ; 'Press space to continue'
                LDA             FoundEsc        ; <ESC> pressed?
                BNE             WeDone          ; escaped out - Quit
                JMP             Begin           ; Let's start Application
                
* Failed Version Check - Finish Alternate Splash Screen & Quit
NoGo            JSR             WriteText       ; routine to write multiple strings
                DA              SplashScreenC   ; address of strings block
                
                JSR             PressAny        ; 'Press space to continue'
                BRA             WeDone2         ; Quit (can't reset TOOrg for < 5.1)
                
* Reset TimeOut Status
WeDone          LDA             #$00            ; for noting proper exit of
                                                ;  TimeOut App - is set to #$01 when
                                                ;   TimeOut organizer is loaded
                STA             TOOrg           ; TimeOut Organizer Loaded?
                
* Quit and Return to AppleWorks
WeDone2         LDA             #$00            ; force re-load of segments -
                STA             OrgInMem        ; to ensure patches aren't re-used
                STA             PrevSeg         ; to ensure patches aren't re-used
```

```
225
226                 RTS                             ; leave TimeOut and go back to AplWorks
227
228 *-------------------------------------------------------
229
230 * Common Text
231 SplashScreenA   STR             '***    TimeOut Edit BASIC    ***'
232                 DB              #$FF            ; centered L-R
233                 DB              #$04            ; row (4)
234                 STR             '- An AppleSoft BASIC Loader, Viewer and Editor -'
235                 DB              #$FF            ; centered L-R
236                 DB              #$07            ; row (7)
237                 STR             'for AppleWorks 5.1'
238                 DB              #$FF            ; centered L-R
239                 DB              #$08            ; row (8)
240                 STR             'by Hugh Hood'
241                 DB              #$FF            ; centered L-R
242                 DB              #$0A            ; row (10)
243                 STR             'version 5.4 (2019)'
244                 DB              #$FF            ; centered L-R
245                 DB              #$0B            ; row (11)
246                 STR             'Application Notes'
247                 DB              #$FF            ; centered L-R
248                 DB              #$0F            ; row (15)
249                 STR             "SSSSSSSSSSSSSSSS"          ; MouseText Mid Bar
250                 DB              #$FF            ; centered L-R
251                 DB              #$10            ; row (16)
252                 DB              #$00            ; end byte
253
254 * Primary Text when Version Check Passes
255 SplashScreenB   STR             'o A BASIC program may be saved as a TEXT file. '
256                 DB              #$FF            ; centered L-R
257                 DB              #$11            ; row (17)
258                 STR             'o To enter the saved TEXT file into memory, Quit'
259                 DB              #$FF            ; centered L-R
260                 DB              #$12            ; row (18)
261                 STR             'AppleWorks, and from the BASIC "]" Prompt,'
262                 DB              #$14            ; column (20)
263                 DB              #$13            ; row (19)
264                 STR             '"EXEC" the saved TEXT file.'
265                 DB              #$14            ; column (20)
266                 DB              #$14            ; row (20)
267                 DB              #$00            ; end byte
268
269 * Alternate Text when Version Check Fails
270 SplashScreenC   STR             '**  This TimeOut Application REQUIRES AppleWorks 5.1  **'
271                 DB              #$FF            ; centered L-R
272                 DB              #$13            ; row (19)
273                 DB              #$00            ; end byte
274
275 *-------------------------------------------------------
276 * Continue with Application
277 *-------------------------------------------------------
278
279 Begin
280                 JSR             MvRightRtn      ; Relocate application above
```

```
                                        ;  SEG $27 (Organizer Main/$2100+),
                                        ;  SEG $28 (File Loader/$5200+), and
                                        ;  SEG $15 (WP Main/$6E00+)
                DA              MoveAdrs        ; Destination / $7E00
                DA              CodeStart       ; Source
                DA              MainCodeEnd-CodeStart         ; Length

                JMP             MoveAdrs        ; Start Application at
                                        ;  Working Address

************************************************************
CodeStart
                ORG             MoveAdrs        ; $7E00

                LDA             #$00            ; force load of Organizer Main segment
                STA             OrgInMem        ; $00 = in file / $01 = in menu /
                                        ;   $40 = in file list

                LDA             #$27+$80        ; Load, don't call Organizer Main
                JSR             CallSeg         ; AW segment loading routine
                LDA             #$28+$80        ; Load, don't call File Loader
                JSR             CallSeg         ; AW segment loading routine

* Clear out any "current" file to make room for new BASIC file
                LDA             DTCurOpen       ; "current" file # (if any)
                JSR             SlotFrCurr      ; routine to save the data about the
                                        ;  "current" file back into its
                                        ;  proper slot (1-12)
                LDA             #$00            ; set "current" file # to equal 0
                STA             DTCurOpen       ; Signal no "current" file yet
                JSR             SlotClrCurr     ; routine to clear (zero out) contents of
                                        ;  "current" file slot from $C56-$C75
                                        ; (DTFName/Path/Type/Status/Picked/Size)

* Draw First (2) Filecards in file selection dialog
                JSR             ClearDA         ; clears display area on screen
                LDA             #$01            ; file card level #1
                JSR             DrawFileCard    ; draw AppleWorks FileCard
                                        ; 'Edit BASIC 5.4'already pushed on
                                        ;  Stack by TimeOut
                JSR             PushStack       ; add a level to Escape road map
                DA              Prompt1         ; 'Select File'
                LDA             #$02            ; file card level #2
                JSR             DrawFileCard    ; draw AppleWorks FileCard
                JSR             PushStack       ; add a level to Escape road map
                DA              PromptQ         ; 'Quit'
                                        ; NOTE: No need to draw file card level
                                        ;       #3 - it will be drawn by FCList
                                        ;       routine called from SEG $28
                                        ;       by AskPathName routine

*-------------------------------------------------------------
* Patches to SEG $27 (Organizer Main) and SEG $28 (File Loader)
*-------------------------------------------------------------

                LDA             #Prompt2        ; Patch Card #3 Stack Msg
```

```
337                 STA             TextLoader+3    ; @ $6369  (LB)
338                 LDA             #>Prompt2       ; 'BASIC files'
339                 STA             TextLoader+4    ; @ $636A (HB)
340
341
342                 LDA             #ParmTable      ; Patch AskPathName ParmTable Address
343                                                 ;  to limited selection to BASIC files
344                 STA             ParmTblPtr      ; $3B28  (LB)
345                 LDA             #>ParmTable
346                 STA             ParmTblPtr+1    ; $3B29 (HB)
347
348                 LDA             #Rename         ; Patch 'Type a new name' routine to
349                                                 ;  default to 'FILENAME.T'
350                 STA             TypeNew+1       ; $4DBF (LB)
351                 LDA             #>Rename
352                 STA             TypeNew+2       ; $4DC0 (HB)
353
354                 LDA             #ChangeName     ; Patch address to new prompt string
355                                                 ;  for TypeNew routine
356                 STA             PromptNew       ; $432F (LB)
357                 LDA             #>ChangeName
358                 STA             PromptNew+1     ; $4330 (HB)
359
360 * Replace/Bypass Remembered Files Routine and Show File as 'New'
361                 LDA             #$A9            ; LDA Instruction
362                 STA             RemFilesRtn     ; ($5D25)
363                 LDA             #$01            ; 'New'
364                 STA             RemFilesRtn+1   ; ($5D25+1)
365                 LDA             #$8D            ; STA Instruction
366                 STA             SetFileStatus   ; Change STZ $0C6C to STA #$01 ('New')
367                 LDA             #$60            ; RTS / Don't remember this file
368                 STA             RemFilesRtn+2   ; ($5D25+2)
369
370 * Set additional RTS points
371                 STA             $4C3B           ; To Return to TimeOut (instead of
372                                                 ;  MainEntry after sucessful file
373                                                 ;     load and CloseAFile)
374
375                 STA             $4D4C           ; To Return to TimeOut (instead of
376                                                 ;  AddMenu after JSR to DTCountRtn
377                                                 ;  ($4D1A) from this TimeOut App and
378                                                 ;  # of files is < 36)
379
380                 STA             TryAgain+3      ; $4DF6
381                                                 ; To Return to TimeOut (instead of
382                                                 ;  (Rename after <ESC> Abort) after
383                                                 ;    CloseAFile patched in below)
384
385 * Patch in Return to TimeOut if we already have 36 files
386                 LDA             #PopAndWrap     ; JMP (not RTS) to PopAndWrap
387                 STA             GoMainEntry+1   ; $4D3C
388                 LDA             #>PopAndWrap    ;
389                 STA             GoMainEntry+2   ; $4D3D
390                                                 ;  (instead of going to MainEntry after
391                                                 ;    JSR to DTCountRtn ($4D1A) from
392                                                 ;     this TimeOut App and # of files
```

```
                                        ;     is 36)
                                        ;  Pop RTS because there is no need to
                                        ;   return and JSR to NewFrOther ($4D99)
                                        ;   File Loader if there is no room
                                        ;   - already 36 files.

* Patch in CloseAFile Routine before RTS to TimeOut after a Rename <ESC>
               LDA           #$20       ; JSR Opcode
               STA           TryAgain   ; $4DF3
               LDA           #$11       ; $2111 (LB) - CloseAFile
               STA           TryAgain+1 ; $4DF4
               LDA           #$21       ; $2111 (HB) - CloseAFile
               STA           TryAgain+2 ; $4DF5

* Patch out direct entry into file after loading
               LDA           #$EA       ; NOP Opcode
               STA           EnterFile  ; $4C35
               STA           EnterFile+1; $4C36
               STA           EnterFile+2; $4C37
                                        ; Defeat entry of file until
                                        ;  AFTER TimeOut is back in
                                        ;  control and is finished
                                        ;  (normally #$88 Resume Application
                                        ;    Call Code for SEG $15 - WP)
                                        ;  (Routine is JMPed to from $4DDC and
                                        ;   includes CloseAFile routine)


* Fix AppleWorks 5.1 bug when zero files are selectable in FCList
*   if making a new file from a Text (or any type) file
*   (should NOT show 'errors loading file' msg if no file was selected)
               LDA           #$EA       ; NOP Opcode
               STA           $3B32      ; Was BEQ Opcode in
               STA           $3B33      ;  AskPathName routine

* Patch in code to add printer option to indent long lines
               LDA           #$4C       ; JMP to our code
               STA           $645B      ;
               LDA           #AddIndent ;
               STA           $645B+1    ;
               LDA           #>AddIndent;
               STA           $645B+2    ;


* Patch in BASIC parser conversion code
               LDA           #$4C       ; JMP to our code
               STA           $6473      ;
               LDA           #Parse     ;
               STA           $6473+1    ;
               LDA           #>Parse    ;
               STA           $6473+2    ;

* Patch in code to handle conversion of single to multiple characters
               LDA           #$4C       ; JMP to our code
               STA           $6469      ;
               LDA           #NewReadMore;
```

```
449                STA             $6469+1         ;
450                LDA             #>NewReadMore   ;
451                STA             $6469+2         ;
452
453 *---------------------------------------------------------------
454 * End of Patches to SEG $27 (Organizer Main) and SEG $28 (File Loader)
455 *---------------------------------------------------------------
456
457                JSR             DTCountRtn      ; ensures no more than 12/36 Files
458
459                JSR             NewFrOther      ; ($4D99) Make WP/Text from 'Other'
460
461                LDA             ErrFlag         ; Any ProDOS Loading Errors?
462                BNE             WrapUp          ;  (don't enter any file)
463
464                LDA             $A57            ; No file was loaded - <ESC> Abort
465                                                ;  (temporary file load status flag)
466                BEQ             WrapUp          ;  (don't enter any file)
467
468 * Make the last file loaded the current file
469                LDA             DTCtOnDesk      ; # of files on Desktop
470                STA             DTCurOpen       ; # of current file
471
472                LDA             DTCurOpen       ; # of current file
473                JSR             Slot2Curr       ; bring last file into current area
474
475 * Reset TimeOut Status
476 WrapUp         LDA             #$00            ; for noting proper exit of
477                                                ;  TimeOut App - is set to #$01 when
478                                                ;   TimeOut organizer is loaded
479                STA             TOOrg           ; TimeOut Organizer Loaded?
480
481 * Quit and Return to AppleWorks
482                LDA             #$00            ; force fresh re-load of patched
483                STA             OrgInMem        ;  segments the next time they are
484                STA             PrevSeg         ;  called
485
486                RTS                             ; leave TimeOut and go back to AplWorks
487
488 *-------------------------------------------------------
489 * Remove (2) RTS Before Quitting
490 PopAndWrap     PLA                             ; pop RTS
491                PLA                             ; pop RTS
492                BRA             WrapUp          ; back to AppleWorks
493
494 *-------------------------------------------------------
495
496 * Various prompt pStrings
497 Prompt1        STR             'Select File'
498 Prompt2        STR             'BASIC Files'
499 PromptQ        STR             'Quit'
500 FCLPrmpt1      STR             'Use Return to select a BASIC file or TAB to change Drives'
501 FCLPrmpt2      STR             'BASIC file'
502 ChangeName     STR             'Type an editing name for this BASIC file:  '
503
504 *-------------------------------------------------------
```

```
505
506 * Parameters for AskPathname/FCList Routine
507 ParmTable      DA              FCLPrmpt1     ; 'Use Return to select BASIC...'
508                DA              FCLPrmpt2     ; BASIC file
509                DB              $C1           ; Pick (1) Only
510                DB              $FC           ; BASIC - FileType for custom description
511                DA              FCLPrmpt2     ; 'BASIC file' - custom description
512                DB              $02           ; (2) Types of files to show
513                DB              $FC           ; show BASIC files in list
514                DB              $0F           ; show Directory files in list
515                DB              $00           ; End Byte
516
517 *--------------------------------------------------------
518
519 * Routine to suggest a '.T' be added to end of BASIC file just loaded
520 Rename         LDX             NewStr        ; Length of File Name
521                CPX             #$0E          ; Less than 14 characters?
522                BCC             :RN
523
524                LDX             #$0D          ; 13 characters
525                STX             NewStr
526 :RN            INX                           ; add 1 for '.'
527                LDA             #'.'          ; '.'
528                STA             NewStr,X
529                INX                           ; add 1 or 'T'
530                LDA             #'T'          ; 'T'
531                STA             NewStr,X
532                STX             NewStr
533                JMP             RealTypeNew   ; ($431E) / Back to regular rename
534
535 *--------------------------------------------------------
536 * BASIC PARSING CODE
537 *--------------------------------------------------------
538
539 *--------------------------------------------------------
540 * First Convert Line #s / Ignore #s 0-1 - Convert #s 2-3
541 *--------------------------------------------------------
542
543 Parse          LDY             InputCount    ; is Zero on initial entry
544                INC             InputCount    ; for next pass through
545                CPY             #$03          ; #s 0-1 are line pointers
546                                              ;  and # 2 is Line # Low Byte
547                BCC             PreNext       ; Ignore; get another
548
549                CPY             #$04          ; #s 3 is Line # High Byte
550                BCS             CheckRest     ; if #4-end is NOT a line number
551
552                STA             $65F6         ; Setup for newWord2Str
553                JSR             newWord2Str   ; convert Hex to Decimal Str
554                DA              $65F5         ; Line # in Decimal Str
555                                              ;  ($65F5 contains 'Y' = $02
556                                              ;    since it is previous character;
557                                              ;   $65F6 contains 'Y' = $03
558                                              ;    since it was read this pass)
559
560                JSR             StrMvRtn      ; Move converted line # string
```

```
561                DA              StrWork9        ; TO
562                DA              StrWork5        ; FROM
563                INC             StrWork9        ; Add 1 for trailing 'space'
564                LDX             StrWork9        ; Length of Line # Str + 'space'
565                LDA             #' '            ; trailing 'space'
566                STA             StrWork9,X      ; store trailing 'space'
567
568 * Pass multiple characters through on single input character
569 StartLoop       INC             MoreFlag        ; Set MoreFlag for multiple chars
570
571                STX             CountFlag       ; Length of String to Priint
572
573                LDA             #$00            ; Initialize Character Counter
574                STA             StringCount
575 LoopPrint      LDX             StringCount
576                INX                             ; Inc 'X' from 1 thru end
577                LDA             StrWork9,X      ; Load Character
578                STX             StringCount     ; Save for next pass
579                CPX             CountFlag       ; Last Character?
580                BCC             Proceed         ; No - Just Pass Character Thru
581
582                DEC             MoreFlag        ; Yes - Unset flag to stop loop &
583                BRA             Proceed         ;  then Pass Character Thru
584
585 *-------------------------------------------------------------
586 * Then Convert Tokens, Pass Characters, Check EOL
587 *-------------------------------------------------------------
588
589
590 CheckRest       CMP             #$00            ; Is char read (in 'A') a null?
591                BEQ             CheckEOL        ; Is it EOL?
592                BMI             Detoken         ; Is it a Token? (>= $80)
593
594                LDX             ControlFlag     ; show as inverse (config option)
595                BEQ             Proceed         ; don't Inverse Controls, Just Pass Thru
596
597                CMP             #$20            ; Is it Control?
598                BCC             ProceedCntrl    ; Pass it as Inverse
599
600                BRA             Proceed         ; NOT EOL or Token or Cntrl - Just Pass Thru
601
602 * Eliminate Nulls at End of File Read
603 CheckEOL        LDA             $65F5           ; Previous Byte Read
604                BEQ             Next            ; Ignore; get another
605
606 * Reset InputCount and Issue Carriage Return for EOL
607                STZ             InputCount      ; begin next line parse
608                JMP             $6533           ; Carriage Return Handler
609
610 * Trap to discard characters after last EOL
611 PreNext         CPY             #$02            ; Is this the third byte on line?
612                BNE             Next            ; No - go get another
613                LDA             $65F5           ; Check Previous Byte Read
614                BNE             Next            ; Line Ptr/second byte was NOT zero
615                STZ             InputCount      ; reset to indicate new line and
616                                                ;  disregard altogether
```

```
617
618 Next            JMP             $6469           ; Ignore; get another char
619
620
621 Proceed         JMP             $647B           ; Back to standard AplWorks filter
622
623 ProceedCntrl    ORA             #$80            ; Make Control Inverse
624                 JMP             $64A5           ; Pass Control as Inverse and JMP
625                                                 ;  over other character filters
626
627
628 Detoken         CMP             #$EB            ; Invalid Token?
629                 BCS             Next            ; Ignore; get another
630                 SEC                             ; Prepare for subtraction
631                 SBC             #$7F            ; Convert Token to Index #
632                 ASL                             ; Multiply by (2)
633                 CLC
634                 TAX                             ; Index into TokenStrs Table
635
636                 LDA             TokenStrs-2,X
637                 STA             TokenAddress
638                 LDA             TokenStrs-1,X
639                 STA             TokenAddress+1
640                 JSR             StrMvRtn
641                 DA              StrWork9        ; TO
642 TokenAddress    DA              TokenHOME       ; FROM (default string)
643                 LDX             StrWork9
644                 BRA             StartLoop
645
646 *--------------------------------------------------------
647 * Counters, Flags and Work Strings
648 *--------------------------------------------------------
649
650 InputCount      DB              $00             ; Position on input line
651
652 MoreFlag        DB              $00             ; More Characters to Pass
653
654 CountFlag       DB              $00             ; # of this Character to Pass
655
656 StrWork9        DS              $0A             ; pString to Pass (10 characters)
657
658 StringCount     DB              $00             ; must save and restore on loops
659
660
661 *--------------------------------------------------------
662 * CONFIGURATION SEG PARAMETERS STORED HERE
663 *--------------------------------------------------------
664
665 ControlFlag     DB              $01             ; show controls (config item - Yes/No)
666
667 IndentCount     DB              $06             ; default to 6 character Indent
668                                                 ;   (config item - choose 0 - 8)
669
670 *--------------------------------------------------------
671 * Handle Instances where (1) Input Character translates to Multiple Characters
672 *  (patched into AppleWorks at $6469)
```

```
673 *-------------------------------------------------------
674
675 NewReadMore     LDA             MoreFlag        ;
676                 BEQ             GetNext         ; no more to print in the string
677
678                 JMP             LoopPrint       ; Finish rest of string
679
680
681 GetNext         JSR             $63C9           ; Process Next Character Read
682                 JMP             $646C           ; Back to Original AplWorks Code
683
684 *-------------------------------------------------------
685 * Allow Indentation of lines that span more than one line.
686 *  (patched into AppleWorks at $645B)
687 *-------------------------------------------------------
688
689 AddIndent       JSR             $638B           ; finish original $645B code
690                 LDA             #$DE            ; Printer token for indent
691                 STA             $B5             ; DLLength - holds printer token
692                 LDA             IndentCount     ; # of spaces to indent (config item)
693                 STA             $B4             ; DLStickies - holds printer token value
694                 JSR             WPPutLine       ; $6E1A - Add Line to Document
695                 JSR             $638B           ; increase line pointer
696                 JMP             $645E           ; back to original code
697
698 *=======================================================
699 * Address table for token pStrings
700
701 TokenStrs       DA              TokenEND
702                 DA              TokenFOR
703                 DA              TokenNEXT
704                 DA              TokenDATA
705                 DA              TokenINPUT
706                 DA              TokenDEL
707                 DA              TokenDIM
708                 DA              TokenREAD
709                 DA              TokenGR
710                 DA              TokenTEXT
711                 DA              TokenPR
712                 DA              TokenIN
713                 DA              TokenCALL
714                 DA              TokenPLOT
715                 DA              TokenHLIN
716                 DA              TokenVLIN
717                 DA              TokenHGR2
718                 DA              TokenHGR
719                 DA              TokenHCOLOR
720                 DA              TokenHPLOT
721                 DA              TokenDRAW
722                 DA              TokenXDRAW
723                 DA              TokenHTAB
724                 DA              TokenHOME
725                 DA              TokenROT
726                 DA              TokenSCALE
727                 DA              TokenSHLOAD
728                 DA              TokenTRACE
```

```
729                 DA              TokenNOTRACE
730                 DA              TokenNORMAL
731                 DA              TokenINVERSE
732                 DA              TokenFLASH
733                 DA              TokenCOLOR
734                 DA              TokenPOP
735                 DA              TokenVTAB
736                 DA              TokenHIMEM:
737                 DA              TokenLOMEM:
738                 DA              TokenONERR
739                 DA              TokenRESUME
740                 DA              TokenRECALL
741                 DA              TokenSTORE
742                 DA              TokenSPEED
743                 DA              TokenLET
744                 DA              TokenGOTO
745                 DA              TokenRUN
746                 DA              TokenIF
747                 DA              TokenRESTORE
748                 DA              TokenAmp
749                 DA              TokenGOSUB
750                 DA              TokenRETURN
751                 DA              TokenREM
752                 DA              TokenSTOP
753                 DA              TokenON
754                 DA              TokenWAIT
755                 DA              TokenLOAD
756                 DA              TokenSAVE
757                 DA              TokenDEF
758                 DA              TokenPOKE
759                 DA              TokenPRINT
760                 DA              TokenCONT
761                 DA              TokenLIST
762                 DA              TokenCLEAR
763                 DA              TokenGET
764                 DA              TokenNEW
765                 DA              TokenTAB
766                 DA              TokenTO
767                 DA              TokenFN
768                 DA              TokenSPC
769                 DA              TokenTHEN
770                 DA              TokenAT
771                 DA              TokenNOT
772                 DA              TokenSTEP
773                 DA              TokenPlus
774                 DA              TokenMinus
775                 DA              TokenStar
776                 DA              TokenBSlash
777                 DA              Token^
778                 DA              TokenAND
779                 DA              TokenOR
780                 DA              TokenGT
781                 DA              TokenEQ
782                 DA              TokenLT
783                 DA              TokenSGN
784                 DA              TokenINT
```

```
                DA              TokenABS
                DA              TokenUSR
                DA              TokenFRE
                DA              TokenSCRN
                DA              TokenPDL
                DA              TokenPOS
                DA              TokenSQR
                DA              TokenRND
                DA              TokenLOG
                DA              TokenEXP
                DA              TokenCOS
                DA              TokenSIN
                DA              TokenTAN
                DA              TokenATN
                DA              TokenPEEK
                DA              TokenLEN
                DA              TokenSTR
                DA              TokenVAL
                DA              TokenASC
                DA              TokenCHR
                DA              TokenLEFT
                DA              TokenRIGHT
                DA              TokenMID

*=========================================================
* Token pStrings

TokenEND        STR             ' END '         ; $80...128
TokenFOR        STR             ' FOR '         ; $81...129
TokenNEXT       STR             ' NEXT '        ; $82...130
TokenDATA       STR             ' DATA '        ; $83...131
TokenINPUT      STR             ' INPUT '       ; $84...132
TokenDEL        STR             ' DEL '         ; $85...133
TokenDIM        STR             ' DIM '         ; $86...134
TokenREAD       STR             ' READ '        ; $87...135
TokenGR         STR             ' GR '          ; $88...136
TokenTEXT       STR             ' TEXT '        ; $89...137
TokenPR         STR             ' PR# '         ; $8A...138
TokenIN         STR             ' IN# '         ; $8B...139
TokenCALL       STR             ' CALL '        ; $8C...140
TokenPLOT       STR             ' PLOT '        ; $8D...141
TokenHLIN       STR             ' HLIN '        ; $8E...142
TokenVLIN       STR             ' VLIN '        ; $8F...143
TokenHGR2       STR             ' HGR2 '        ; $90...144
TokenHGR        STR             ' HGR '         ; $91...145
TokenHCOLOR     STR             ' HCOLOR= '     ; $92...146
TokenHPLOT      STR             ' HPLOT '       ; $93...147
TokenDRAW       STR             ' DRAW '        ; $94...148
TokenXDRAW      STR             ' XDRAW '       ; $95...149
TokenHTAB       STR             ' HTAB '        ; $96...150
TokenHOME       STR             ' HOME '        ; $97...151
TokenROT        STR             ' ROT= '        ; $98...152
TokenSCALE      STR             ' SCALE= '      ; $99...153
TokenSHLOAD     STR             ' SHLOAD '      ; $9A...154
TokenTRACE      STR             ' TRACE '       ; $9B...155
TokenNOTRACE    STR             ' NOTRACE '     ; $9C...156
```

```
841 TokenNORMAL     STR             ' NORMAL '      ; $9D...157
842 TokenINVERSE    STR             ' INVERSE '     ; $9E...158
843 TokenFLASH      STR             ' FLASH '       ; $9F...159
844 TokenCOLOR      STR             ' COLOR= '      ; $A0...160
845 TokenPOP        STR             ' POP '         ; $A1...161
846 TokenVTAB       STR             ' VTAB '        ; $A2...162
847 TokenHIMEM:     STR             ' HIMEM: '      ; $A3...163
848 TokenLOMEM:     STR             ' LOMEM: '      ; $A4...164
849 TokenONERR      STR             ' ONERR '       ; $A5...165
850 TokenRESUME     STR             ' RESUME '      ; $A6...166
851 TokenRECALL     STR             ' RECALL '      ; $A7...167
852 TokenSTORE      STR             ' STORE '       ; $A8...168
853 TokenSPEED      STR             ' SPEED= '      ; $A9...169
854 TokenLET        STR             ' LET '         ; $AA...170
855 TokenGOTO       STR             ' GOTO '        ; $AB...171
856 TokenRUN        STR             ' RUN '         ; $AC...172
857 TokenIF         STR             ' IF '          ; $AD...173
858 TokenRESTORE    STR             ' RESTORE '     ; $AE...174
859 TokenAmp        STR             ' & '           ; $AF...175
860 TokenGOSUB      STR             ' GOSUB '       ; $B0...176
861 TokenRETURN     STR             ' RETURN '      ; $B1...177
862 TokenREM        STR             ' REM '         ; $B2...178
863 TokenSTOP       STR             ' STOP '        ; $B3...179
864 TokenON         STR             ' ON '          ; $B4...180
865 TokenWAIT       STR             ' WAIT '        ; $B5...181
866 TokenLOAD       STR             ' LOAD '        ; $B6...182
867 TokenSAVE       STR             ' SAVE '        ; $B7...183
868 TokenDEF        STR             ' DEF '         ; $B8...184
869 TokenPOKE       STR             ' POKE '        ; $B9...185
870 TokenPRINT      STR             ' PRINT '       ; $BA...186
871 TokenCONT       STR             ' CONT '        ; $BB...187
872 TokenLIST       STR             ' LIST '        ; $BC...188
873 TokenCLEAR      STR             ' CLEAR '       ; $BD...189
874 TokenGET        STR             ' GET '         ; $BE...190
875 TokenNEW        STR             ' NEW '         ; $BF...191
876 TokenTAB        STR             ' TAB( '        ; $C0...192
877 TokenTO         STR             ' TO '          ; $C1...193
878 TokenFN         STR             ' FN '          ; $C2...194
879 TokenSPC        STR             ' SPC( '        ; $C3...195
880 TokenTHEN       STR             ' THEN '        ; $C4...196
881 TokenAT         STR             ' AT '          ; $C5...197
882 TokenNOT        STR             ' NOT '         ; $C6...198
883 TokenSTEP       STR             ' STEP '        ; $C7...199
884 TokenPlus       STR             ' + '           ; $C8...200
885 TokenMinus      STR             ' - '           ; $C9...201
886 TokenStar       STR             ' * '           ; $CA...202
887 TokenBSlash     STR             ' / '           ; $CB...203
888 Token^          STR             ' ^ '           ; $CC...204
889 TokenAND        STR             ' AND '         ; $CD...205
890 TokenOR         STR             ' OR '          ; $CE...206
891 TokenGT         STR             ' > '           ; $CF...207
892 TokenEQ         STR             ' = '           ; $D0...208
893 TokenLT         STR             ' < '           ; $D1...209
894 TokenSGN        STR             ' SGN '         ; $D2...210
895 TokenINT        STR             ' INT '         ; $D3...211
896 TokenABS        STR             ' ABS '         ; $D4...212
```

```
TokenUSR        STR             ' USR '         ; $D5...213
TokenFRE        STR             ' FRE '         ; $D6...214
TokenSCRN       STR             ' SCRN( '       ; $D7...215
TokenPDL        STR             ' PDL '         ; $D8...216
TokenPOS        STR             ' POS '         ; $D9...217
TokenSQR        STR             ' SQR '         ; $DA...218
TokenRND        STR             ' RND '         ; $DB...219
TokenLOG        STR             ' LOG '         ; $DC...220
TokenEXP        STR             ' EXP '         ; $DD...221
TokenCOS        STR             ' COS '         ; $DE...222
TokenSIN        STR             ' SIN '         ; $DF...223
TokenTAN        STR             ' TAN '         ; $E0...224
TokenATN        STR             ' ATN '         ; $E1...225
TokenPEEK       STR             ' PEEK '        ; $E2...226
TokenLEN        STR             ' LEN '         ; $E3...227
TokenSTR        STR             ' STR$ '        ; $E4...228
TokenVAL        STR             ' VAL '         ; $E5...229
TokenASC        STR             ' ASC '         ; $E6...230
TokenCHR        STR             ' CHR$ '        ; $E7...231
TokenLEFT       STR             ' LEFT$ '       ; $E8...232
TokenRIGHT      STR             ' RIGHT$ '      ; $E9...233
TokenMID        STR             ' MID$ '        ; $EA...234

                DB              $00             ; END OF TOKEN NAME TABLE

*=========================================================

                ORG

MainCodeEnd


*-----------------------------------------------------------
* Configuration Segment
*-----------------------------------------------------------

Config
* Configuration screen strings - Titles
                STR             'TimeOut Edit BASIC Configuration'
                DB              #$FF            ; Column (centered)
                DB              #$06            ; Row #6
                STR             "SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS"           ; MT Mid Bar
                DB              $FF             ; Column (centered)
                DB              $07             ; Row #7

                DB              #$00            ; end of screen strings

*-----------------------------------------------------------
* Configuration Options

:1              STR             'Show Control Characters in INVERSE'        ; Menu Item
                DB              $0D             ; Column (Question)
                DB              $0B             ; Row (Question)
                DB              $40             ; Column (Answer)
                DB              $0B             ; Row (Answer)
                DB              $03             ; Length (Answer)
```

```
                DA              :2-*            ; Bytes to Next Question
                DB              #$05            ; Yes/No Type Question
                DB              #$01            ; SEG #1 to Patch
                DA              ControlFlag-MoveAdrs+CodeStart-Start          ; Offset

:2              STR             '# of Characters to Indent Long Lines'        ; Menu Item
                DB              $0D             ; Column (Question)
                DB              $0D             ; Row (Question)
                DB              $40             ; Column (Answer)
                DB              $0D             ; Row (Answer)
                DB              $03             ; Length (Answer)
                DA              :3-*
                DB              #$03            ; Byte/Integer Type Question
                DB              #$01            ; SEG #1 to Patch
                DA              IndentCount-MoveAdrs+CodeStart-Start          ; Offset
                DB              $00             ; Minimum accepted number
                DB              $08             ; Maximum accepted number

:3
                DB              #$00            ; End of Config Parameters
*-------------------------------------------------------
                ORG

CodeEnd
                LST             off

                TYP             $06             ; BIN file (also need $2100 AuxType)
                SAV             TO.EDITBASIC54
```