Displaylist in ACTION!#

General Information

Author: William T. Colburn
Language: ACTION!
Compiler/Interpreter: ACTION!
Published: February, 1987

This program makes use of the routine ALLOCATE.ACT from the OSS Action! Tool Kit. ALLOCATE.ACT is copyrighted by OSS and is NOT in the public domain. Therefore, ALLOCATE.ACT was not included with DLIST2.ACT by the Author, and must NOT be added to it for public distribution by any other user of this demonstration program. The author of this routine refuses to accept responsibility for this type of unethical action by any users of my demonstration program.

```
MODULE ;DLIST2.ACT

;************************************
;*This program is in the public   *
;*domain and may not be sold by   *
;*anyone for any reason.  It was     *
;*written by:                             *
;*       William T. Colburn         *
;*       in February, 1987.         *
;* CI$: 72337,322  GEnie:W.T.COLBURN*
;*Please enjoy this demonstration   *
;*program and share it with others! *
;*Please keep this header with the  *
;*program listing when you do so.   *
;************************************
;*              * NOTICE *              *
;*This program makes use of the   *
;*routine ALLOCATE.ACT from the OSS *
;*Action! Tool Kit.  ALLOCATE.ACT is*
;*copywrited by OSS and is NOT in   *
;*the public domain.  Therefore,     *
;*ALLOCATE.ACT was not included with*
;*DLIST2.ACT by the Author, and   *
;*must NOT be added to it for public*
;*distribution by any other user of *
;*this demonstration program.        *
;*The author of this routine refuses*
;*to accept responsibility for this *
;*type of unethical action by any   *
;*users of my demonstration program.*
;************************************

CARD EndProg ;required for ALLOCATE.ACT

; You  must  do a 'SET EndProg=*'
; from the monitor  after  compiling,
; but  before  running this program!

INCLUDE "D8:ALLOCATE.ACT"; from the Action! Tool Kit

MODULE ; My gloabl variables here.

BYTE ARRAY dlist=  ; display list!
```

```
                [
                 $22 ; length (34 bytes)
                 $70 $70 $70 ;24 overscan lines
                 $42 $00 $00 ;load address of static here.
                 $06 $06      ;two lines of Gr.1
                 $42 $00 $00 ;load savmsc+200 here.
                 $02 $02 $02 ;22 lines Gr. 0
                 $02 $02 $02
                 $02 $02 $02
                 $02 $02 $02
                 $02 $02 $02
                 $02 $02 $02
                 $02 $02
                 $41 $00 $00 ;load address of dlist+1 here.
                ]
BYTE ARRAY static ; static 80 byte display, allocated with Alloc() from Action! tool k

PROC dsply_list()
    CARD savmsc=$58, ;contains low address of screen display
          dlist_vector=$230, ; points to the display list
          old_savmsc=[0], ; save the savmsc here.
          temp_card=[0] ; temporary variable!
    BYTE dma=559,  ; antic chip on/off address
          crsinh=752, ;cursor on/off address
          loop ; loop counter
    BYTE POINTER dlist_ptr,; pointer to display list array.
                     save_dlist_ptr,
                     static_ptr ; pointer to static
    dlist_ptr=dlist
    save_dlist_ptr=dlist
    save_dlist_ptr==+1
    static=Alloc(81); allocate 81 bytes for 'static'.
    FOR loop=1 TO 80
        DO
            static(loop)=0
        OD
    static(0)=80 ; set length of string
    static_ptr=static ; set pointer
    static_ptr==+1 ; point to entry #1.
    Graphics(0)
    old_savmsc=savmsc ; save start of screen adress
    dlist_ptr==+5
    dlist_ptr^=static_ptr-((static_ptr RSH 8) LSH 8)
    dlist_ptr==+1
    dlist_ptr^=static_ptr RSH 8 ;divide by 256!
    dlist_ptr==+4
    temp_card=old_savmsc+120
    dlist_ptr^=temp_card-((temp_card RSH 8) LSH 8)
    dlist_ptr==+1
    dlist_ptr^=temp_card RSH 8
    dlist_ptr==+22
    dlist_ptr^=save_dlist_ptr-((save_dlist_ptr RSH 8) LSH 8)
    dlist_ptr==+1
    dlist_ptr^=save_dlist_ptr RSH 8 ;divide by 256!
    dma=0 ; turn off the antic chip
    dlist_vector=save_dlist_ptr; install the dlist vector
    savmsc=old_savmsc; reset the screen starting address
    dma=34 ; turn on the antic chip
;**crsinh=1 ; kill cursor
```

```
RETURN

MODULE ; for user.

PROC Main()
    BYTE lmargn=$52,
          dma=559, Answer=[0]
    lmargn=3
    PutE()
    PutE()
    PrintE("Did you do a:  SET EndProg=* ")
    PrintE("from the monitor after compiling")
    PrintE("but before running this program?}}")
    Print("Respond Y or N...?")
    Answer=GetD(7)
    Put(Answer)
    IF Answer='Y OR Answer='y THEN
        Print("}")
    ELSE
        RETURN
    FI
    AllocInit(0); from ALLOCATE.ACT
    dsply_list(); install display list
    SetColor(2,10,3); pick your color
    SAssign(static,"Colburn-Systems",09,32)
    SAssign(static,"professional",45,56)
    SAssign(static,"display-list-manager",61,80)
    Print("}") ; clear the screen
    Position(3,4)
    PrintE("See?  This is the display list!")
    Position(3,7)
    PrintE("This screen has a static display")
    PrintE("on the first three lines of the")
    PrintE("screen.  The rest scrolls.")
    PrintE("The tricky part of all this is")
    PrintE("that the characters in 'static'")
    PrintE("which you want to display must be")
    PrintE("in the internal Atari code because")
    PrintE("they won't be translated!")
    PrintE("<Or some such pain in the neck.>")
    PrintE("To see scrolling, type 'E' and then")
    PrintE("press RETURN to go back to the")
    PrintE("editor.  The display list will")
    PrintE("remain in place until you return")
    PrintE("to the monitor or hit RESET.")
    lmargn=0
RETURN
```