

DOUBLE WINDOW MENU

A smart menu program that lets you organize your programs by category

Having to scroll through long lists of disk directories, looking for the program you want? Organize your programs with Double Window Menu. You can divide the programs on each disk into categories, and see only the programs associated with the category you've selected.

USING THE PROGRAM

Using Double Window Menu couldn't be easier. The top of the screen is used to identify the series of programs being displayed and provides instructions on how to advance the cursor bar and select the program desired. Below the title, the screen is divided by what looks like an oversized T. See Figure 1. On the left side is the primary menu, and the right side contains the secondary menu.

As you move the cursor bar from one category to another in the primary menu area with the arrow keys, the secondary menu is automatically updated to display the programs associated with the selected primary menu item. Once you have chosen the category you want, press Return and the cursor jumps to the right window, where the secondary menu is. Again use the arrow keys to select the desired program, and press the Return key to run the program. If you press the Escape key while using the secondary menu, you will be transferred back to the primary menu.

Since the program in Listing 1 is just a demonstration, the program you select is not actually run; to use Double Window Menu with your own programs, see the Customization section.

ENTERING THE PROGRAM

To key in the program, type in Listing 1 and save it with the command

```
SAVE DUBL.WINDO.MENU
```

You may want to name the program STARTUP if you want this program to be run when you boot your ProDOS disk. DOS 3.3 users may want to initialize their disk with this program as their HELLO program. Remember that Double Window Menu program is not meant to be a general purpose menu but must be set up with the data statements appropriate for the filenames on your disk.

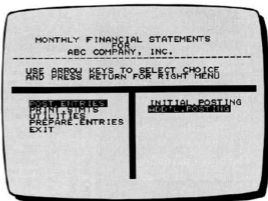


Figure 1: Double Window Menu in Action

HOW THE PROGRAM WORKS

When the program is run, the data statements at the end of the program are READ into a two-dimensional array, OPTNS(NC,NR), where NC represents the number of columns in your table of choices and NR represents the number of rows or choices in the primary menu.

Once the initialization phase is complete, the screen displays the primary menu on the left, the right side displays the program names associated with the first category in the primary menu.

The program then scans the keyboard for an arrow or Escape or Return keypress (lines 150-260). The program will ignore any input other than those keys. If an arrow keypress is detected, the cursor bar will select another category in the primary menu and the secondary menu will display the choices applicable to that category. Here is where the windows come in.

By POKing memory locations 32 and 33, you control which part of the screen you want to be active. The rest of the screen remains frozen. After the highlighted item in the primary window is changed, the right window is activated and allows a fast update of the new choices. Then the primary menu regains control.

When you press Return, the secondary menu becomes active. Now when you press the arrow keys, the cursor on the left side of the screen remains frozen while the secondary cursor moves. If you press Escape, you will be returned to the primary menu. A Return keypress will attempt to run the program currently highlighted in the secondary menu.

CUSTOMIZATION

To set up your own choices you will have to organize your data statements so they will be properly read into the array named OPTNS(). Column zero of the array represents the primary menu choices; column one represents programs associated with the first category in the primary menu; column two represents programs associated with category two, and so on.

Remember to make the last item in your primary menu EXIT. The program will exit to BASIC when you choose the last item in the primary menu.

When entering the data statements, you should use quotation marks and pad each name with spaces so that the cursor bar will have the same length when it passes from one item to the next.

When you use Double Window Menu with your own files, change line 520 to read

```
520 TEXT : HOME : PRINT OPTNS(CL,K) : PRINT : PRINT
      CHR$(4) : "RUN ",OPTNS(CL,K) : END
```

If you are using ProDOS, you can replace "RUN" with "--". This will allow you to BRUN binary files, EXEC text files, and boot system programs such as AppleWorks.

LISTING 1: DUBL.WINDO.MENU

```
37 10 REM *****
38 20 REM * DUBL.WINDO.MENU *
B9 30 REM * BY FRED V. AMUNDSEN *
AE 40 REM * COPYRIGHT (C) 1988 *
CB 50 REM * BY MICROSPARC, INC. *
24 60 REM * CONCORD, MA 01742 *
45 70 REM *****
27 80 TEXT : HOME : GOSUB 560 : GOTO 480
41 90 REM SHOW CHOICES
0C 100 GOSUB 450 : HOME : FOR KK = 1 TO C(K) : PRIN
      T OPTNS(K, KK) : NEXT KK : GOSUB 460 : RETURN
A7 110 REM GET CHOICE
D8 120 V = VT + 2 : NORMAL
17 130 LET K = 1
88 140 HTAB 1 : VTAB V : INVERSE : PRINT OPTNS(CL, K
      ) : NORMAL : IF CL = 0 THEN GOSUB 100
3D 150 LET C = PEEK (KYBD) : IF C < 128 THEN 150
01 160 POKE STROBE, 0
78 170 IF (C = 14) AND CL = 0 AND K = MAX) THEN
      TEXT : HOME : END
A8 180 IF C = 141 AND CL = 0 THEN : VTAB (21) :
      CALL - 958 : PRINT 'PRESS ESC TO' : PRINT
      'RETURN TO' : PRINT 'THE LEFT WINDOW' :
      RETURN
DB 190 IF C = 155 AND CL > 0 THEN POP : CL = 0 : MA
      X = C(0) : GOTO 490
EE 200 IF C = 141 THEN RETURN
20 210 IF C = 149 OR C = 138 OR C = 136 OR C = 13
      9 THEN 230
8F 220 GOTO 150
C9 230 VTAB V : HTAB HT : PRINT OPTNS(CL, K)
53 240 IF C = 149 OR C = 138 THEN V = V + 1 : K = K
      + 1 : IF K > MAX THEN V = VT + 2 : K = 1
16 250 IF C = 136 OR C = 139 THEN V = V - 1 : K = K
      - 1 : IF K = 0 THEN V = VT + 2 + MAX - 1 : K
      = MAX
41 260 GOTO 140
D2 270 REM SHOW HEADINGS
F8 280 HOME : PRINT TAB(20 - LEN(L1$) / 2) : L1
      $ : PRINT TAB(20 - LEN(L2$) / 2) : L2$ :
      PRINT TAB(20 - LEN(L3$) / 2) : L3$ : FOR
      K = 1 TO 40 : PRINT "-": NEXT
88 290 PRINT
21 300 PRINT TAB(4) : "USE ARROW KEYS TO SELECT C
      HOICE" : PRINT TAB(4) : "AND PRESS RETURN FO
      R RIGHT MENU."
C2 310 BLS = " " : INVERSE : REM BLANK SPACE
78 320 VTAB 9
```

```
99 330 FOR K = 1 TO 40 : PRINT BLS : NEXT
1D 340 FOR K = 1 TO 14
1B 350 HTAB 21 : PRINT BLS
62 360 NEXT K
F4 370 VTAB 1
23 380 NORMAL
70 390 RETURN
2B 400 REM SHOW MENU
3E 410 V = 11
92 420 FOR K = 1 TO MAX
D2 430 HTAB HT : VTAB V : PRINT OPTNS(CL, K) : V = V
      + 1 : NEXT K
20 440 RETURN
D3 450 POKE 32,22 : POKE 33,18 : POKE 34,10 : RETURN
      : REM SETUP RIGHT WINDOW
EF 460 POKE 32,2 : POKE 33,17 : POKE 34,10 : RETURN
      : REM SETUP LEFT WINDOW
40 470 REM MAINLINE
E6 480 GOSUB 280
83 490 CL = 0 : VT = 9 : HT = 1 : GOSUB 460 : HOME :
      GOSUB 410 : GOSUB 120
2C 500 GOSUB 450 : CL = K : MAX = C(K) : HOME : FOR KK
      = 1 TO MAX : PRINT OPTNS(CL, KK) : NEXT
84 510 GOSUB 120 : REM GET CHOICE
8A 520 TEXT : HOME : PRINT OPTNS(CL, K) : VTAB 23 :
      PRINT "IN A REAL APPLICATION," : PRINT "YOU
      WOULD NOW BE" : PRINT "RUNNING " : OPTNS(CL,
      K) : "" : PRINT "PRESS RETURN TO REST
      ART/ESCAPE TO QUIT " : POKE - 16368, 0 : GET
      AS
01 530 IF AS < > CHR$(27) THEN RUN
55 540 END
FC 550 REM INITIALIZATION
F2 560 PRINT CHR$(21) : TEXT : HOME
2F 570 D$ = CHR$(4)
AF 580 L1$ = "MONTHLY FINANCIAL STATEMENTS"
33 590 L2$ = "FOR"
A1 600 L3$ = "ABC COMPANY, INC."
49 610 REM CHANGE NEXT LINE FOR YOUR MENU
02 620 NC = 5 : NR = 20 : REM NUMBER OF COLUMNS AND
      ROWS
A9 630 LET KYBD = - 16384 : LET STROBE = - 16368
      : HT = 3
49 640 DIM OPTNS(NC, NR)
DA 650 DIM C(NC)
44 660 FOR J = 0 TO NC
57 670 READ C(J)
CA 680 FOR K = 1 TO C(J)
85 690 READ OPTNS(J, K)
40 700 NEXT K : NEXT J
E0 710 LET MAX = C(0)
C2 720 RETURN
75 730 REM DATA STATEMENTS FOR FILES ON DISK
17 740 DATA 5 : REM COLUMN ZERO
35 750 DATA "PREPARE ENTRIES"
66 760 DATA "POST ENTRIES"
3D 770 DATA "PRINT STMTS"
88 780 DATA "UTILITIES"
A2 790 DATA "EXIT"
8A 800 DATA 2 : REM COLUMN ONE
E9 810 DATA "STANDARD ENTRIES"
B6 820 DATA "SPECIAL ENTRIES"
AB 830 DATA 2 : REM REM COLUMN TWO
CE 840 DATA "INITIAL POSTING"
59 850 DATA "ADD L POSTING"
DA 860 DATA 4 : REM COLUMN THREE
9B 870 DATA "ONE PAGE STMT"
52 880 DATA "PRINT LEDGER"
C2 890 DATA "PRINT ENTRIES"
58 900 DATA "DIVISION ANALYSIS"
CB 910 DATA 13 : REM COLUMN FOUR
FF 920 DATA "ADD ACCOUNTS"
6B 930 DATA "NEW ENTRIES"
CF 940 DATA "CLEAR ACCOUNTS"
5E 950 DATA "DIRECTORY ADJUST"
DA 960 DATA "RESTART"
9C 970 DATA "GENERAL LEDGER"
C8 980 DATA "MERGE ENTRIES"
E7 990 DATA "TRANSFER DATA"
53 1000 DATA "COPY FILES"
9D 1010 DATA "COPY DISK"
61 1020 DATA "DOS UP"
74 1030 DATA "INITIALIZE DISK"
0D 1040 DATA "INSTRUCTIONS"
25 1050 DATA 1 : REM COLUMN FIVE
9E 1060 DATA "EXIT TO BASIC" : REM REQUIRED LAST
      STATEMENT
```

TOTAL: 314F
END OF LISTING 1