U9. Other Utility Commands

ViewIt also supports several miscellaneous commands that deal with setting the cursor, posting events, and idle procs used with printing.

Name   Number   Parameters & Variables used

ChgCur   121   a,fCursor

Resets the cursor to that designated by parameter a where 0 = Arrow, 1 = IBeam, 2 = Cross, 3 = Plus, 4 = Watch, and all other values are interpreted as CURS or crsr (color cursor) resource ID numbers. The fCursor variable is also set equal to -1 to force the cursor to be updated when control is finally returned to FaceIt or ViewIt.

PstEvt   122   a,b,c,d

Creates a new event and puts it into the event queue where the event record will be defined as what = a, message = b, where = c, modifiers = d, and the when field will be set to the current time. PstEvt offers greater control over the posted event than the corresponding toolbox call PostEvent.

When operating under FaceIt, FaceSt, or ViewIt, event 13 (a = 13) is used to pass messages between modules and the main program. PstEvt places these events in a private queue that is given priority over the Macintosh raw event queue. Events from this private queue are then seen by the program as menu or "pseudo-menu" events (via DoLoop, MdlWnd, GetMsg).

PstEvt can also be used by programs to "fake" user actions or to post special messages back to a program's event loop. *To fake the selection of a program menu item, pass,*

a = 13
b = 0
c = menuID (-> uMenuID)
d = menu item number (-> uMenuItem)

To fake a hit in a ViewIt window, pass,

a = 13
b = FWND ID (-> uMenuID)
c = 0
d = item ID (-> uMenuItem)

and set any other fRec variables that are expected to return information with window hits (wiHit, wvHit, wcHit, etc.). To fake direct messages from modules, pass,

a = 13
b = FCMD ID (-> uMenuID)
c = module info (-> uResult)
d = message number (-> uMenuItem)

Note that the effects of most of these calls will depend on the current program context (i.e., which window is active or control is selected), so you may need to first bring windows forward and/or select controls.

To post a special message back to your program's event loop, simply pass a value of b (-> uMenuID) that your program can distinguish from other pseudo-menu events. Choose a value of b that is outside of the range of FCMD IDs (1100-7499), and is not equal to any label ID, menuID, or FWND ID (a number > 7499 is a good choice). Parameters c and d can be used to pass additional information with the message, where the value of c will be returned in uResult and d in uMenuItem.

NOTE: The value returned in uMenuItem gets temporarily stored in the 2-byte "modifiers" field of an event record, so don't try to use it to pass 4 bytes of information. uResult, on the other hand, gets stored in the 4-byte "where" field, so you can use this to pass addresses, handles, etc.

PstNot   124   a,uResult

This command posts a "Notice" type message to FaceWare modules. It is typically used with the FaceSt (FaceStub) module to post "Save Documents" (a = 4096) and "Clean System" (a = 32) Notices to modules before quitting the program. *In the case of the "Save Documents" message, uResult will return with a non-zero value if the user cancels the operation.*

SetIdl   131   a,b,c,uResult

EndIdl   132

SetIdl sets the "pIdleProc" field of the designated print record equal to ViewIt's own background idle procedure that supports a "rotating" cursor, use of the command key to pause printing, and command-period to halt printing, where

a = the print record handle (type "THPrint")

b = first of 5 successive CURS resources to use for cursor rotation, where the first cursor is displayed during a pause and remaining are displayed in succession during printing   (if b = 0, then ViewIt's own cursors are used (b = 1220))

c = tick delay between cursor change (governs speed of rotation)

Call SetIdl just before calling PrOpenDoc.   If the user terminates printing by pressing command-period, then uResult is set equal to -1 and the appropriate printing error is posted.   The idle procedure allocates a temporary block and stores its address at $A80.   When finished printing, call EndIdl to clear the "pIdleProc" field of the print record, dispose of the temporary block, and restore the previous value at $A80.   Be careful not to call EndIdl without calling SetIdl.