U3. Window Utility Commands

Several ViewIt commands are supported that deal with the managing of window-related attributes such as style, title, zoom state, order, and position.

Name   Number   Parameters & Variables used
SetFSS   183   a,b,c,d,uString,uResult
Resets the font, size, and style fields of the designated window or port.   Passing a value of -1 for b, c, or d causes ViewIt to ignore that parameter.   If the port's font, size, or style is changed, then ViewIt returns uResult ≠ 0.
*   a = target window or port*
*   0 = front modal or active modeless window*
*   1 = current port*
*   other = WindowPtr or GrafPtr*
*   b = font number*
*   or b = -2 = get font name from uString*
*   or b = address of a Pascal string containing font name*
*   c = font size (12 pt. used if c = 0)*
*   d = style = sum of following: 0 = Plain,*
*   1 = Bold, 2 = Italic, 4 = Underline, 8 = Outline,*
*   16 = Shadow, 32 = Condensed, 64 = Extended*
NOTE: SetFSS does not change the appearance of text in an existing window.   It only changes the appearance of new text drawn by the program after calling SetFSS (i.e., it has the same effect as calling TextFont, TextSize, and TextFace).

SndBhd   250   a,b,d
*   Moves a window to a new location in the window list (i.e., above or below some other window).   SndBhd also calls ShowWindow to show the window if it is not already visible.   Avoid using SndBhd if a modal window is the front window.*
*   a = WindowPtr of window to move*
*   b = where to move the window in window list*
*   1 = send to back (below all other windows)*
*   0 = bring to front, but below any FaceIt floating palettes*
*   -1 = bring to front using "SelectWindow"*
*   -2 = bring to front without calling "SelectWindow"*
*   other = WindowPtr of a window to send window a behind*
*   d = update mode (same use as in ViewIt's DoUpdt)*
The b = -1 or -2 options are primarily for FaceWare's internal use.   The most common use of SndBhd is to bring a modeless window to the front:
*   FaceIt(nil,SndBhd,ord(myWindow),0,0,-1);*
where d = -1 updates window contents and other stuff.

GetNam   251   a,b,c,uName
*   Returns all or part of a window's current title in uName (if c = 0), or in the Pascal string whose address is given by c, where a is the window's WindowPtr, and b designates which part of the title to return: 0 = entire title, 1 = window name only, 2 = file name only (the text between "" and ""), or 3 = file name if not empty, else window name.*
NOTE: This command is primarily used by modules that make use of a window's title to display a file name.

SetNam   252   a,b,c,uName
*   Resets all or part of a window's current title to uName (if c = 0), or to the Pascal string whose address is given by c, where a is the window's WindowPtr, and b designates which part of the title to reset: 0 = entire window title, 1 = window name only, and 2 = file name only (the text in the title that is between "" and "").*
NOTE: This command is primarily used by modules that make use of a window's title to display a file

name.

MovRec   254   a,b,c,d,uRect
MovAlt   255   a,b,c,d
MovDlg   256   a,b,c,d
MovWin   257   a,b,c,d
Moves a rectangle to the screen position designated by the parameters b, c, and d.   For MovRec, uRect is adjusted.   For MovAlt, MovDlg, and MovWin, the boundsRect of the corresponding ALRT, DLOG, or WIND resource (in memory, not on disk) whose resource ID is equal to a is adjusted.   The resulting rectangle is always kept within the visible screen area.   MovRec can also be used to move an existing window by passing a WindowPtr in parameter a.
   a = resource ID of ALRT, DLOG, or WIND (not MovRec)
     or window procID† (MovRec only)
     or WindowPtr of window to move (MovRec only)
   b = what device (screen), window, or position to use as basis for movement
     -2 =   c and d are global coordinates
     -1 = all devices
     0 = main device (screen with menu bar)
     1 = deepest device (greatest bits per pixel)
     2 = front window††
     3 = current cursor location
     4 = current device that contains cursor
     5 = window w/ selected control or active modeless or front modal window††
     6 = global rectangle passed in uRect
     7 = active modeless or front modal window††
     other = WindowPtr of an existing window
   c = vertical placement†††
     0 = no change
     1 = centered
     2 = at top
     3 = at bottom
     4 = 1/3 of distance from top to bottom
     other = pixel offset from top
   d = horizontal placement†††
     0 = no change
     1 = centered
     2 = at left
     3 = at right
     other = pixel offset from left
A typical use of MovDlg, for example, would be to adjust the boundsRect of a DLOG resource before opening a new dialog window,
 FaceIt(nil,MovDlg,1000,0,1,1);
which would center DLOG 1000 on the main screen.   The MovRec command, on the other hand, could be used to move a hidden window into position before it is shown:
 FaceIt(nil,MovRec,ord(myWindow),7,1,1);
which would center "myWindow" above the front modal or active modeless window.   Another common use of MovRec is to force a window to be completely visible when part of it might be out of view:
 FaceIt(nil,MovRec,ord(myWindow),-1,0,0);
which will only move the window if part of it is off-screen.
† When using MovRec to adjust the position of uRect, the procID or "window definition ID" passed in parameter a is used to account for the title bar and border area of a window that might be formed from uRect.   Use a = procID = 2, for example, to minimize the correction for title bar and border since this corresponds to a window with no title bar and a minimal border.
†† Passing b = 2 bases movement on the front window, even if it is a floating window.   If positioning one window above another, it will usually make more sense to pass b = 5 which uses either the window with the selected control, the active modeless window, or, if one or more modal windows are open, the top modal window.

DrgWnd   259   a,b,c,d,uRect

   Drags and, if d ≠ 0, selects the designated window.   This command does not call "DragWindow" or "SelectWindow", making it useful in cases where these traps may have been patched and the caller wishes to drag and select the window in a controlled manner (ViewIt uses DrgWnd, for example, when in editing mode).
   a = WindowPtr (or 0 for front modal or active modeless)
   b = starting point of drag (global coordinates)
   c = address of bounding rect (or 0 = use fDragRect)
   d = selection and update mode
      0 = don't select after drag
      other = update mode defined by ViewIt's DoUpdt

GetStd   301   a,uRect

   Returns in uRect the maximum standard state zoom rect for the window designated by WindowPtr a. The rectangle is adjusted to properly fit within whichever screen device contains the top, left corner of the content area of window a.

SetStd   302   a,uRect

   Uses the contents of uRect to reset the standard zoom state rectangle of the window designated by WindowPtr a. SetStd is typically called by modules after "fine tuning" the uRect returned by GetStd.

SavDlg   356   a,b,c,d
SavWin   357   a,b,c,d

   Updates and then saves a DLOG or WIND resource back to disk where,
   a = WindowPtr to get boundsRect rectangle from
   b = resource ID of DLOG (SavDlg) or WIND (SavWin)
   c = value to place in refCon field of resource
      (or -1 to use window's current refCon value)
   d = value (0 = F, 1 = T) to put in visible field of resource
      (or -1 to use window's current visibility)
   NOTE: This command has been made largely obsolete by the new ViewIt module which uses custom FWND resources.