FaceIt Demo Source Code Notes

The following compiler-independent notes correspond to the "C#" comments in the "fDemoXY" source code.   Read these notes while examining a printed copy of the source.

C1. "fDemoXY" provides example code that corresponds to the topics discussed in the FaceIt Guide which is found in the "ViewIt Help" window opened from the File icon menu when in ViewIt's edit mode.   Also see the "Startup" topics in the ViewIt Help window for a description of basic resource set up and initialization tasks performed at the start of the program.

C2. The FaceStorXY file defines FaceIt, ViewIt, and UtilIt command constants and the fRec shared record used by all FaceWare modules.   The FaceProcXY file defines the "FaceIt" dispatching procedure and other supporting procedures.   All communication with FaceWare modules is made through this FaceIt dispatching procedure using the command and variable names defined in "Stor" files such as FaceStorXY.

C3. A good way to deal with program-specific resources during development is to keep them in a separate res file (a "temporary resource file").   FaceWare facilitates use of such a file by allowing you to pass the name of this file in the fRec variable uName when calling DoInit.   In this program the name being passed is "fDemo.Rsrc" which the FaceIt dispatching procedure will attempt to open if it cannot find the necessary resources within the program file.   Note that no harm is done by leaving this assignment in your program source after combining the temporary resource file with the program file.

C4. Before making use of any FaceWare module, DoInit must be called.   DoInit is described in "Initializations" in the startup topics, and in the "Commands" topic in the FaceIt Guide.   In this program we are asking that the cursor be changed to a deck of cards whenever control is returned to the main program (add 1 to a), that control be returned to the program whenever the active window is changed (2), and that control be returned if the user attempts to open or print files "from the Finder" (512).   This explains why a = 515 = 1 + 2 + 512 is passed.

C5. The "Editor" and "Clipboard" modeless windows that magically appear are automatically loaded by FaceIt when DoInit is called based on instructions found in the STR# 1000 ("FaceInit") resource:

```
 1200,20,1010
 1200,20,1020
```

where the first opens a modeless ViewIt 2.0 window (baseID 1200, versID 20) using FWND 1010 which contains a TextCt text editing control, and the second opens a modeless ViewIt window using FWND 1020 which contains a ScrapCt clipboard display control.   See "Initializations" in the FaceIt Guide for further info.

C6. This "Beeps & Notes" modeless window is opened by calling NewWnd with a = 1030 = FWND ID, and b = 1 indicating that the window is to be modeless (vs. modal).   See "Windows" in the ViewIt Guide for further info.

C7. The main program event loop is an infinite loop that responds to program-specific events not handled by other modules.   The first statement simply passes control to FaceIt by calling DoLoop.   FaceIt then keeps control until an event occurs that belongs to the main program.   See "The Main Loop" in the FaceIt Guide for further info.

C8. As described in "The Main Loop" topic, the main program receives two types of events: pseudo-menu or menu events.   In this program a case (or if...then...else) block based on uMenuID and uMenuItem is used to process events.

C9. The first item in the Apple menu (uMenuItem = 1, uMenuID = 101) causes the program to open a simple alert via UtilIt's ShoStr command.   ShoStr and other string display commands make it easy to quickly display such messages in any Font, Size, Style, and Color.

C10. uMenuIDs 105, 106, and 107 correspond to the three menus in this program that contain "Beep Once", "Beep Twice", and "Beep Thrice" as their first items:
   105 = main menu bar menu entitled "fDemo"
   106 = hier. menu attached to "Hierarchical" item
   107 = hier. picture palette menu attached to "Hier. Palette" item and also popped up from item in window

C11. Selection of one of the "Beep" items in these menus causes the program to respond by beeping once, twice, or thrice (= the value of uMenuItem).

C12. Item #8 (uMenuItem = 8) from the "fDemo" menu (uMenuID = 105) is used to illustrate topics discussed in "Background Processing" in the FaceIt Guide: support for background processing and switching, and the monitoring of events without calling Get/WaitNextEvent.

C13. The UtilIt command ShoAlt is used to open ALRT 1010 centered horizontally and vertically on the main screen (b = 0, c = 1, d = 1).   (A modal ViewIt window could also be used as an alert.)

C14. If "Cancel" was not chosen (uResult = mode > 1), then the program enters an endless loop that beeps once every three seconds.

C15. If demonstrating background processing/switching (mode = 2), then GetNextEvent is called each time thru the loop to check for update events (fEvent.what = 6), switching events (fEvent.what = 15), and other events of interest (in this case, auto-key events = fEvent.what = 5).   If update or switching events are found, they are passed on to FaceIt to handle via DoEvnt.   If an auto-key event is found, then the loop is exited.

C16. If demonstrating the monitoring of events without calling Get/WaitNextEvent (mode = 3), then the toolbox call "GetKeys" is used to check the status of the keyboard.   If, in this example, the bits corresponding to the Option and Command keys are both set, then the loop is exited.   When using "BitTst" to test bits, the following bit offsets apply: 48 = Command, 61 = Option, 62 = Caps Lock, 63 = Shift, and 40 = Period key.

C17. ViewIt returns control with a pseudo-menu event (menuID = 1030 = FWND ID, uMenuItem = item # pressed) when a button is pressed in the modeless "Beeps & Notes" window.   In this window there are 3 buttons that correspond to beeping once, twice, or thrice (wcHit = 1, 2, or 3).   This window also contains an enabled picture control that also generates a pseudo-menu event when clicked (returning wcHit = 4).

   Each time the picture is clicked, the program responds by using UtilIt's PopMen to pop up the picture palette that was previously auto-installed by FaceIt on DoInit.   It pops up the menu directly above the picture in the window by specifying the proper local coordinates when calling PopMen.   PopMen returns with uMenuID and uMenuItem reset to values corresponding to the user's choice from the pop-up menu.   Note that this pop-up menu is the same non-main menu as that displayed via the "Hier. Palette" item in the "fDemo" main menu.

C18. The next case is another example of responding to a pseudo-menu event. The event being processed is one that we asked to be informed of when calling DoInit: a change in the identity of the active window. This event returns with uMenuID = 1100 (the baseID of FaceIt module), and uMenuItem = 2 (to distinguish it from other pseudo menu events posted by FaceIt). Knowing when the active window changes is useful when you have program menu items that are "context sensitive". In this example we simply change the text of the last item in the "fDemo" menu to track the type of window that is active. The active window type is easily identified by the baseID of the window-driving module (stored in fActiveID), and the ID of the FWND used to open the window (stored in fActiveResID).

C19. The last case is another example of responding to a pseudo-menu event. The event being processed is also one that we asked to be informed of when calling DoInit: an attempt by the user to open or print files "from the Finder". This event returns with uMenuID = 1100 (the baseID of FaceIt), uMenuItem = 512, uResult = 1 (open) or 2 (print), uString = file type, uName = file name, and the current directory set to that containing the file needing to be opened or printed. To keep the example simple, this code only opens TEXT files from the Finder into a single, existing window with a TextCt text-editing control. The command GetCtl is used to get the control's control handle which is then used when executing TextCt's OpnCTxt command to open the file. Also note how the file name returned in uName is preserved across the GetCtl command since calls to the FaceIt procedure do not preserve "u" variables. See "Finder Resources" in the FaceIt Guide for more info.