Simpler Source

ViewIt is powerful but, unlike other tools, does not make you pay for this power with complex source code. At the same time this does not mean that you sacrifice any flexibility when using ViewIt since you can choose to intercept every event and message that is passed to a ViewIt control.

The simplest programs that use ViewIt also make use of the FaceIt module to handle the main menu bar and other program-wide features. FaceIt-based programs require just a few lines of code (Pascal source is shown but other languages are similar):

```
FaceIt(nil,DoInit,0,0,0,0);     initialize stuff
repeat
  FaceIt(nil,DoLoop,0,0,0,0);   handle events
until false;
```

where the single "FaceIt" procedure is used to call all FaceWare modules, and the first parameter "nil" (= zero) indicates that ViewIt, FaceIt, or UtilIt is being called.

The source for this demo program is slightly more complicated than that shown above due to the few items in its windows that require a response from the program. Hitting the Why ViewIt? button in the "Modeless ViewIt Window", for example, causes the program to execute the following code to open this "Why ViewIt?" modal window:

```
FaceIt(nil,NewWnd,1003,0,0,0);     open window
repeat
  FaceIt(nil,MdlWnd,1003,0,0,0);   run window
until (wcHit = 1);
FaceIt(nil,EndWnd,1003,0,0,0);     close window
```

where "wcHit" is the number of a control in this window (wcHit = 1 = "OK" button).

How can so little code be required to open and run this window? The trick is that ViewIt and ViewIt controls understand how to handle most events that occur in windows, allowing the program code to be reduced to that needed to handle program-specific events. Other program and dialog managers also do this, but none does it as well as ViewIt.

The scrollable view #2, for example, "knows" what to do when its scroll bar is clicked. The menus in the bar above this help text ("File", "Edit", "Text", & "Topics") contain standard items that "know" what to do when selected. And ViewIt "knows" what to do when the grow or zoom boxes are used to resize the window. None of this default behavior requires a single line of program code to support it.