

Off-Line Documentation template:          Flexible Scrolling model

### **1) Operational Goals**

Provide a flexible scrolling model in DinkClass which can be used for the types of scrolling I know about (image-painting and data control)

### **2) Fundamental, "key", or cornerstone architectural requirements (POSTMORTEM)**

- Need controls controlling the values of some variables
- Need some implementation of the Focus on window/content which is modifiable
- Need a scroll bar setting / thumb position validation mechanism
- Need a modifiable way of defining scroll step sizes
- Need to maintain the ranges the scroll bars can go
- Keep things comprehensible, by not using the control manager data structures as storage for application variables (like what Dan Weston did in his ScrollDoc class, boo!)

### **3) Model of the implementation fulfilling these key requirements (POSTMORTEM)**

- Scroll bars control DWindow members fVOffset and fHOffset.
- The window subclass needs to define what the values of fVOffset and fHOffset imply ( and hence what FocusOnContent means)
- The scroll bars are defined in terms of the window dimensions, the offsets, and the scroll bar ranges.
- The Scroll bars need to have their ranges validated whenever the window changes size or the scroll ranges get changed. (validation is not needed on a simple scroll)
- It is up to the subclasses to define how the scroll ranges are specified.
- It is up to the subclasses to define what Scroll Contents means.

### **4) Impact/scope of the implementation on the existing body of code (POSTMORTEM)**

- DWindow has the offsets and virtual definitions of what FocusOnWindow and FocusOnContent mean
- DScrollWindow has the controls, control ranges, validation methods, and the UI.
- DScrollWindow uses Scroll Contents as the bottleneck procedure for actually doing the scroll operator.
- DScrollWindow has a lot of comments through out the .h and .c files

### **5) Coding notes (gotchas, warnings, process thoughts, items to revisited later...)**

- Hopefully you will not have to mess with the ActionProc, providing the feed back of the page and button scrolls. If you just HAVE to mess with it then perhaps its time to take that static member out of the class declaration and put the code in a more isolated place, to make application specific modifications easier.
- One key concept important to scrolling is that the ranges defined by the window represent the range of the VISIBLE area and the ranges held within the scroll bars are the ranges required such that the existing window's content region can access the entire range at its current size.
- to put it another way, the scroll Bar ranges are the ranges needed to move the content rect such that all of the entire visible area can be seen.

**6) Testing notes( bug types, what made a bug hard to fix, what could have been done to catch it sooner....)**

**7) Process notes ( what process did you follow, could it be improved)**