README

Visible View

This program is used as a teaching aid in the NeXT Developer Camp. Its purpose
is to demonstrate the features of the View class, a class that provides access
to PostScript, knows how to draw itself on the screen and handles events.

To use the program, bring up any of the Tool panels and watch the results of
your manipulations of the View's various components.

> · The Frame panelÐthe location of the View in its superview's
> coordinates space.
> · The Coordinates panelÐits own coordinates space, the area in which
> you draw.
> · The Appearance panelÐadd a grid to the View, showing its own
> coordinate space, to composite in an image as the background
> (compositing is a very fast way to paint existing bitmaps into a
> View), make the superview transparent.
> · The Statistics panelÐa readout of the values of several important

instance variables of the View.

The Status panel gives on-going information about the events the View is receiving as well as other actions the View might be taking in response to an event.

You can mouse down in the View to test its response to events. It will beep on mouse down and display the location of the mouse in the view's coordinates on mouse up, even when you drag the mouse outside the View's frame.

NOTES ON DIFFERENCES BETWEEN VISIBLEVIEW-02 AND VISIBLEVIEW-03
In VisibleView-02, when running MallocDebug, you'll notice a leak everytime a string is written out to the status window.
This is being caused by the call to NXCopyStringBuffer(), which allocates a copy of the string. However, this string never gets deallocated.

In the VisibleView-03, this method was replaced with NXUniqueStringNoCopy(), which resolves the leak problem. See the ToolsNotes Performance CheckList for more details.

Left as additional exercises:
Use UserPaths to improve Postscript drawing performance.