

# CompositeLab

CompositeLab is a simple application useful for interactively demonstrating NeXTSTEP compositing modes. Two separate images, source and destination, whose colors (and opacity) are chosen through the use of NXColorWells & the NXColorPanel, are composited together using a compositing mode specified by the user. The images are kept instances of NXImage. One subclass of View, CompositeView, manages all target/actiUCXmethods and drawing.

User can select three colors: Background, source, and destination. Background color is the color painted in the view in all three panes. The source image is first cleared to transparent, then painted using the source color (unless the source image comes from an image file, in which case the source color is ignored). The destination image is created the same way, by first clearing the image and then painting in the destination color. Then the destination is composited to the result image using the Copy mode, and the source is composited on top using the user-specified compositing mode. Finally all three images are displayed in the view using the Sover mode. Thus you get to see the three images as seen above the specified background color.

NXImages are used with the `drawMethod:inObject:` method. Three methods (`drawSource:`, `drawDestination:`, and `drawResult:`) provide the PostScript to draw the three images. The NXImages call these methods and cache the drawing whenever they need to; this happens when they are composited the first time and whenever they are composited after the contents change (indicated by a call to the `recache` method). If CompositeLab allowed printing, these methods would also be called during printing, to produce the image at the full resolution of the printer rather than at the resolution they were cached at in the window server.

CompositeLab also demonstrates how to accept colors and images dragged in through the use of interapplication drag mechanism. In addition to the minimally required methods (`draggingEntered:` and `performDragOperation:`), `CompositeView` overrides `draggingUpdated:` and `draggingExited:` to give the user feedback as he or she drags a color over the view without dropping it.