

Q. The CompositeLab example in /NextDeveloper/Examples/AppKit sometimes hangs when I drag in non-standard image types for which filters need to be run. Typically it wakes up after a minute or so without having accepted the image; however, if I drag the same image in again, it accepts it without a problem.

A. During a drag session, an application should not make calls to Workspace or call methods and functions that would require contacting Workspace. Launching filter services, applications, obtaining icons for files, sliding images, and methods in the Workspace protocol fall in this category. Making such a request will cause the application to hang until either the request or the drag service times out; the results can be unpredictable and often both the request and the drag operation will fail.

In the case of CompositeLab, the application asks `NIImage` to open an file from the **performDragOperation:** method. If the image requires a filter and the filter needs to be launched, this request hangs CompositeLab. However, the same request succeeds

the second time as the filter will have been launched by the Workspace after the first attempt times out.

The easiest workaround is to open the image in the **concludeDragOperation:** instead of the **performDragOperation:**. This latter method is invoked while the drag session is still in process, so requests to Workspace hang. The **concludeDragOperation:** method, on the other hand, is called after the drag session is terminated, so the application may contact Workspace during that time.

The following changes will make CompositeLab work correctly. Replace the existing **performDragOperation:** method with the following two methods. Note that although this new **performDragOperation:** doesn't perform the operation, it still validates the dragged data. This is important—the return value of this method determines whether Workspace causes the dragged file(s) to fly back to the source, indicating success or failure to the user.

```
- (BOOL)performDragOperation:(id <NXDraggingInfo>)sender
{
    return ([self draggingUpdated:sender] == NX_DragOperationNone) ? NO : YES;
}

- concludeDragOperation:(id <NXDraggingInfo>)sender
{
    Pasteboard *pboard = [sender draggingPasteboard];

    if (includesType([pboard types], NXColorPboardType)) {
        [self doColorDrag:sender];
        [sourceColorWell setColor:sourceColor];
        [destColorWell setColor:destColor];
        [backColorWell setColor:backgroundColor];
    } else {
        (void)[self changeCustomImageTo:[NXImage allocFromZone:[self zone]]
initFromPasteboard:pboard]];
    }
    return self;
}
```

```
}
```

If an application wishes to actually open and display images during the drag session, it should ask `UIImage` if the image can be handled directly or requires a filter service, and open the image only if it falls into the former category. Of course, actually opening and displaying the image during the drag session isn't advisable in the first place as this operation might take a while.

QA878

Valid for 3.0