

Transparent Background

Posted At : October 6, 2007 8:33 AM | Posted By : Jon Hartmann

Related Categories: ColdFusion

One of the big frustrations about CF8 image effects, is the lack of any way to get transparent writes or backgrounds. While most people think of transparent as just one more color, when you look at how Java handles pixels, you'll understand why Adobe didn't take the time to do this themselves.

Java images are made of up "integer pixels", a numeric value in which there are 4 bytes, representing alpha channels, red channel, green channel, and blue channel, in that order. Java image writing functions (on which CF8's functions are just a simplified interface) take in a color to write with, and "paint" that color over the pixels it is designated to effect.

Painting with a fully opaque color looks like it is replacing the painted over pixel with the new color, but is really calculating a new color value based on the original color and opacity and the new color and opacity. This means if you try and paint a 0 opacity color, its like layering clear paint over a color; you only see the original color.

So in order for Adobe to make "transparent" paint, they would have to recreate all the image functions in Java, except instead of doing the color calculation to paint, they would instead have to set the color directly... thats a lot of work!

Creating Transparent Backgrounds

Ok, what you have really been waiting to see. This code will only work if your server has not be configured to protect CF internal Java objects, so if your server has this on, you're on your own.

```
<cffunction name="ImageNewTransparent" returntype="any" output="false" hint="Returns an image with a transparent background.">
    <cfargument name="width" type="numeric" required="false" default="200" />
    <cfargument name="height" type="numeric" required="false" default="200" />

    <cfset var type_int_argb = CreateObject("java", "java.awt.image.BufferedImage").TYPE_INT_ARGB />
    <cfset var BufferedImage = CreateObject("java", "java.awt.image.BufferedImage").init(arguments.width, arguments.height,
type_int_argb) />

    <cfreturn CreateObject("java", "coldfusion.image.Image").init(BufferedImage) />
</cffunction>
```

This can be tested with code similar to the following:

```
<cfset myImage = ImageNewTransparent() />
<cfset ImageDrawLine(myImage, 0, 0, 100, 100) />
<div style="background: red;">
    <cfimage action="writeToBrowser" source="myImage" />
</div>
```