

# Transparent Rounded Corners for the Masses

Posted At : April 4, 2009 1:50 AM | Posted By : Jon Hartmann

Related Categories: ColdFusion

Ok, so a late night of coding here... I should have been in bed like 2+ hours ago, but its all worth it because I've got it working: fully transparent rounded corners on your images. Just hit "more" to see the code.

## The Code

```
<cffunction name="ImageRoundedCorners" output="false" returntype="any">
    <cfargument name="image" type="any" required="true" />
    <cfargument name="archeight" type="numeric" required="false" default="0" />
    <cfargument name="arcwidth" type="numeric" required="false" default="0" />

    <cfset var local = StructNew() />

    <!-- Initialize java classes -->
    <cfset local.BufferedImage = CreateObject("java", "java.awt.image.BufferedImage") />
    <cfset local.RoundRectangle2D = CreateObject("java", "java.awt.geom.RoundRectangle2D$Float") />

    <!-- Get Information to pull width and height -->
    <cfset local.info = ImageInfo(arguments.image) />

    <!-- Create a new transparent image the correct size -->
    <cfset local.newImage = local.BufferedImage.init(
        local.info.width,
        local.info.height,
        local.BufferedImage.TYPE_INT_ARGB
    ) />

    <!-- Create the rounded corner clipping mask -->
    <cfset local.clippingMask = local.RoundRectangle2D.init(
        JavaCast("float", 0),
        JavaCast("float", 0),
        JavaCast("float", local.info.width),
        JavaCast("float", local.info.height),
        JavaCast("float", arguments.archeight),
        JavaCast("float", arguments.arcwidth)
    ) />

    <!-- Get the graphics object -->
    <cfset local.graphics = local.newImage.createGraphics() />

    <!-- Set the clipping mask -->
    <cfset local.graphics.clip(local.clippingMask) />

    <!-- Draw the original image over the old one -->
    <cfset local.graphics.drawImage(
        ImageGetBufferedImage(arguments.image),
        0,
        0,
        JavaCast("null", "")
    ) />

    <!-- Convert the image and return it -->
    <cfreturn ImageNew(local.newImage) />
</cffunction>
```

## The Usage

```
<cfset myImage = ImageNew("warbird.jpg") />  
<cfset myImage = ImageRoundedCorners(myImage, 25, 25) />  
<cfimage action="writetobrowser" source="#myImage#" />
```

Its that easy.

---

## Update

Yeah, the code that has been pointed out over at [cfSearching](#) definitely blows mine out of the water... I didn't think I was the first, but man, that post must have been hiding all of the times I searched for similar solutions. Rather than reinventing the wheel, I'll just point you to go get the code from over there :)