My Introduction to Image Manipulation in ColdFusion8

Posted At : October 16, 2008 1:44 PM | Posted By : Charlie Griefer
Related Categories: ColdFusion

One of our client web sites allows users to generate custom marketing materials and save them as PDF files. We had a request yesterday to allow users to save these PDFs as images so that they could embed them directly into e-mails, as opposed to sending as attachments. Sounded like a job for some of the new image manipulation capabilities of ColdFusion 8. I'd not yet had the opportunity to play around with this new functionality, so I was looking forward to it.

At first, I thought it was going to be as easy as using <cfpdf> with the "thumbnail" action. The hitch that I ran into was that some of the PDFs are multiple pages, and <cfpdf action="thumbnail"> generates a thumbnail image for each page. So the challenge was combining each of those individual thumbnails into a single image file. That's where I really got to get my hands dirty with the new built-in image functions. Here's what I came up with (breakdown follows the code):

```
<cffunction name="pdfToImage" output="false" returntype="void">
 <cfargument name="imageFormat" type="string" required="false" default="png" />
 <cfargument name="pdfPath" type="string" required="true" />
 <cfargument name="pdfFileName" type="string" required="true" />

 <!--- create a struct to hold function-specific vars --->
 <cfset var local = structNew() />
 <cfset var idx  = "" />

 <cfset local.customPDFInfo = "" /> <!--- info about the custom-generated PDF... need # of pages --->
 <cfset local.pdfImageInfo  = "" /> <!--- info about the thumbnail images generated... need height/width --->
 <cfset local.newPDFImage   = "" /> <!--- a new image to paste the thumbnails into... stitches together a single image --->
 <cfset local.startingY     = 0 />  <!--- Y-axis coordinate on the new image... where to paste each thumbnail --->

 <!--- get info about the PDF (returns a struct, interested in 'TotalPages') --->
 <cfpdf action="getInfo" source="#arguments.pdfPath##arguments.pdfFileName#.pdf" name="local.customPDFInfo" />
 <!--- create the images (creates one image for each page in the PDF) --->
 <cfpdf action="thumbnail" source="#arguments.pdfPath##arguments.pdfFileName#.pdf" destination="#arguments.pdfPath#" scale="100" format="#arguments.imageFormat#" resolution="high" overwrite="true" />

 <!--- create cfimage objects for each thumbnail created from the PDF pages --->
 <cfloop from="1" to="#local.customPDFInfo.totalPages#" index="idx">
  <cfimage source="#arguments.pdfPath##arguments.pdfFileName#_page_#idx#.#arguments.imageFormat#" name="local.pdfImg#idx#" />
 </cfloop>

 <!--- need the height and width to set the new image dimensions --->
 <cfset local.pdfImageInfo       = imageInfo(local.pdfImg1) />
 <cfset local.newPDFImage        = imageNew('',local.pdfImageInfo.width,local.pdfImageInfo.height * local.customPDFInfo.totalPages) />

 <cfloop from="1" to="#local.customPDFInfo.totalPages#" index="idx">
  <cfset imagePaste(local.newPDFImage, local['pdfImg' & idx], 0, local.startingY) />
  <cfset local.startingY = local.startingY + local.pdfImageInfo.height />
 </cfloop>

 <cfset imageWrite(local.newPDFImage, '#request.workingPdfPath##arguments.pdfFileName#.#arguments.imageFormat#') />
</cffunction>
```

So, what's going on there? Glad you asked.

The function takes 3 arguments... image format (can be tiff, png, or jpg, defaults to png), the path to the PDF, and the PDF file name. After declaring a few function-specific variables, the first functional line is

```
<cfpdf action="getInfo" source="#arguments.pdfPath##arguments.pdfFileName#.pdf" name="local.customPDFInfo" />
```

This returns a structure containing information about the PDF to be converted. What I'm interested in is the 'TotalPages' value. I'll be using this later to loop over the generated thumbnail images and "stitch" them into a final image to be returned to the user.

Once I've got that information, the initial conversion from PDF to image is as simple as

```
<cfpdf action="thumbnail" source="#arguments.pdfPath##arguments.pdfFileName#.pdf" destination="#arguments.pdfPath#" scale="100" format="#arguments.imageFormat#" resolution="high" overwrite="true" />
```

As previously mentioned, this generates an image file (in this case, png) for each page in the PDF document. If there were a way to tell the tag to generate a single image, I'd pretty much be done at this point. But as far as I'm able to tell (and please feel free to correct me if I'm wrong), I'm stuck with multiple images (assuming a multiple-page PDF). So I need to take these individual images and reconstruct a single image.

The first <cfloop>:

```
<cfloop from="1" to="#local.customPDFInfo.totalPages#" index="idx">
 <cfimage source="#arguments.pdfPath##arguments.pdfFileName#_page_#idx#.#arguments.imageFormat#" name="local.pdfImg#idx#" />
</cfloop>
```

loops for the total number of pages in the PDF and creates a ColdFusion image "object" for each of the corresponding thumbnail images that was generated via the <cfpdf> tag.

The next step makes use of 2 of the new built in image manipulation functions. imageInfo() and imageNew().

```
<cfset local.pdfImageInfo = imageInfo(local.pdfImg1) />
```

This gets me a structure that contains information about the first thumbnail that was generated. All I need is the height and width, which I expect to be consistent between all of the generated thumbnails, so I'm safe in passing only the first one to the image.

```
<cfset local.newPDFImage = imageNew('',local.pdfImageInfo.width,local.pdfImageInfo.height * local.customPDFInfo.totalPages) />
```

This creates a new ColdFusion image. The first argument is empty, as I'm not providing a source. For now, it's just a placeholder. It's the image into which each of the previously generated thumbnails will be "stitched". Using the values that I obtained from the imageInfo() function, I set the width, and the height. In this case, the height will be the height of the thumbnail multiplied by the total number of pages.

The penultimate step is to loop over each of the previously generated thumbnails, pasting them into the **newPDFImage** image that I created via the imageNew() function:

```
<cfloop from="1" to="#local.customPDFInfo.totalPages#" index="idx">
 <cfset imagePaste(local.newPDFImage, local['pdfImg' & idx], 0, local.startingY) />
 <cfset local.startingY = local.startingY + local.pdfImageInfo.height />
</cfloop>
```

Here I make use of yet another of the new built in functions, imagePaste(). For each iteration of the loop, I call imagePaste() telling it which image to paste into (the newly created **newPDFImage** image), what image to paste (for each iteration of the loop, one of the previously generated thumbnail images), the x-axis coordinate (will always be zero), and the y-axis coordinate. The y-axis coordinate will change for each iteration of the loop, so I've declared the variable **startingY**, initialized to zero. Then for each loop iteration, I increment it by the thumbnail height so the next iteration will paste the new thumbnail at the appropriate position.

The final step is to write the image to disk using imageWrite().

Again, this is my first foray into image manipulation via ColdFusion. If there's anything that could have been done differently and/or more efficiently, please feel free to drop me a comment below.

New Tags/Functions covered today:

- <cfpdf action="getInfo|thumbnail">
- <cfimage>
- imageInfo()
- imageNew()
- imagePaste()
- imageWrite()

All in all, a pretty good day

UPDATE: As per Todd Sharp's request, I've put up a demo at http://www.amcomtech.net/labs/pdfToImage/. Please don't break my server :)