

Filling out PDF Forms in ColdFusion

How to use the cfdocument tag and some Dreamweaver magic



By Jeff Houser

A few years back, when CF5 was still in the beta stages, a client wanted to be able to fill out PDF forms through their web browser and save, and print, the filled out PDF form.

After experimenting with a few free methods, such as the use of FDF files, I stumbled upon the ActivePDF toolkit. ActivePDF provides a series of tools for PDF manipulation. The toolkit is a COM object that can be used to fill out PDF forms. It did the job wonderfully. I spent more time creating the PDF form with Acrobat than I did creating the ColdFusion markup to allow people to fill out the form via the web.

Now we can jump forward to more recent times, I recently had the opportunity to revisit the project. ColdFusion MX does not deal with COM as well as CF5 did, so we decided to look into other options. CFMX7 has some built in PDF capabilities, so investigating them seemed like an obvious choice. Although CFMX7 does not deal with PDF forms, there is an alternative method that worked well for our purposes. We used the cfdocument tag and some Dreamweaver magic to implement our solution. I'm going to explain that method to you today.

Prepping the Form

You can do this with any PDF form. For example purposes, I'm going to use the US copyright form PA, downloadable from there web site, here: <http://www.copyright.gov/forms/formpa.pdf>. You'll want to extract the PDF as an image. I was able to open it in Adobe Photoshop and save each page as an 8.5 x 11 image. If you don't have access to Photoshop, there are also various low cost converter programs that will convert a PDF's contents to an image. Two such programs are ReaConverter pro from ReaSoft, or VeryPDF's PDF to Image Converter. They are located at <http://www.reasoft.com/products/reaconverter/> and <http://www.verypdf.com/pdf2tif/index.htm> respectively.

Now you have your form as an image. What next? I know this will be a tough move for many of you hard-core developers out there, but next we are going to use Dreamweaver in design mode. These are the steps:

1. Create a new page. I named mine "formip.cfm" for a "form input processing" page.
2. Expand the property panel if it is not already expanded. If it is not displayed, you can display it using "Properties" from the

window menu. Click the "Page Properties..." button. If you don't see this button in the property panel, click on the body tag and it should show up.

3. You should see a window like the one shown in Figure 1. For the category select appearance. For the background image, click browse and select the form that you
4. Click OK to save your changes. Now move to the design view by clicking the design button. You should see your form in the background of the page. Depending on the size of your screen, it will probably tile but for our purposes it doesn't matter.
5. Display the insert bar in Dreamweaver if it is not already displayed. You can do this from the select "insert" from the window menu. Click the layout tab of the insert bar. The insert bar is shown in Figure 3. Roll your mouse over the third button and it should say "Draw Layer". This will allow you to draw a layer anywhere on the screen. Behind the scenes it will create a div layer, positioned using absolute position. Click this button.
6. You will now be able to draw the layer anywhere on the page just by clicking and dragging. Create a layer for each line of the form that you want to fill out.
7. Adjust the layer properties to give it a background color that is equal to the background color of your form. This will probably be white. Each layer will contain one form field, surrounded by pound signs, so that ColdFusion knows to process it. For the title of this work field, I would name it "#form.titleofwork#". You can add these specifics in either the code view of the design view, but it may be easier in the design view to tell which layer goes to which line of the form.

Here's a sample of the finished template. For brevity, I only included two fields in the sample template (Title of Work, and Previous Title):

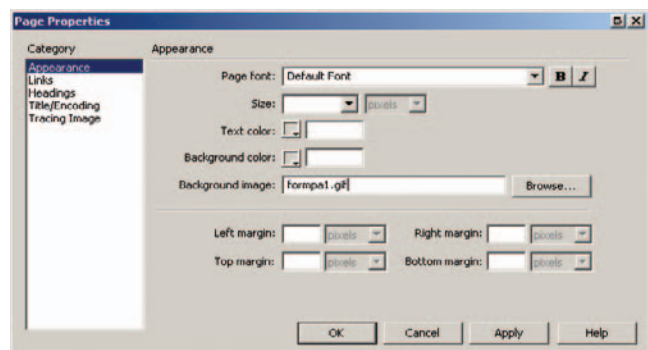


Figure 1: The Page Properties Dialog

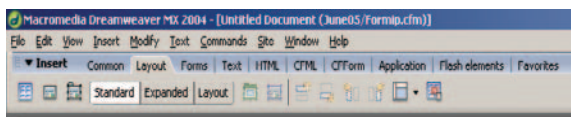


Figure 2: Menu



Figure 3: Insert Bar

```
<cfdocument format="PDF">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/
loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/
html; charset=iso-8859-1">
<title>Copyright Form PA PDF </title>
<style type="text/css">
<!--
  body {
    background-image: url(formpa1.gif);
  }
-->
</style>

</head>

<cfoutput>
  <body>
    <div id="Layer1" style="position:absolute;
left:86px; top:199px; width:502px; height:27px;
border:1px none #000000; z-index:1">
      #form.TitleOfWork#
    </div>
    <div id="Layer2" style="position:absolute;
left:86px; top:243px; width:503px; height:22px;
border:1px none #000000; z-index:2">
      #form.PreviousOrAlternate#
    </div>
  </body>
</html>
</cfoutput>
</cfdocument>
```

Using the cfdocument tag

Create a new page in Dreamweaver. This one can be done all in code view if you want. I named this page, formi.cfm (for "form input"). The code is as follows:

```
<form action="formip.cfm" method="post">
```

```
Title Of Work: <input
type="text"
name="TitleOfWork"><br>
Previous Or Alternate: <input
type="text" name="PreviousOrAlte
rnate"><br>
<input type="submit">
</form>
```

It contains an HTML form with two input boxes, one for the Title of the Work and one for the Previous or Alternate title. The form

posts onto the formip.cfm page that you created in the previous section of this article. You probably aren't finding anything new to you in this template.


Switch back to your "formip.cfm" template. You'll notice that the whole generated html is wrapped in a cfdocument tag. This tag tells ColdFusion to generate a PDF or FlashPaper document. Here are some of the attributes to the cfdocument tag:

- **Format:** Format is a required attribute. The two accepted values for this are FlashPaper and PDF, depending on the type of document you want to create. In our example, we set it to PDF.
- **FileName:** The filename attribute is an optional attribute. If you specify a filename then ColdFusion will generate your PDF or Flashpaper document and save it as a file. If you do not specify a filename then the resulting file will be streamed to the browser. In this example, the code streamed the resulting PDF to the browser.
- **Overwrite:** The overwrite attribute is an optional attribute that is used in conjunction with the filename attribute. It accepts a Boolean value, and if set to yes then the file will be overwritten if it already exists. If set to no, then an error will be thrown if the file exists.
- **Fontembed:** FontEmbed is an optional attribute. It is a Boolean attribute, so accepts either yes or no as values. If set to yes, then all fonts are embedded in the finished document. If set to no, then fonts are not embedded.
- **BackgroundVisible:** The backgroundvisible attribute is an optional attribute that specifies whether the background prints when the user prints the document.

- **Name:** The name attribute is an optional attribute. If you want to store the PDF or FlashPaper document as a variable within ColdFusion then you'll use the name attribute. In this example, you were streaming the results to the browser and this was not used.

For a complete list of all attributes supported by the cfdocument tag, you should review the documentation located at <http://livedocs.macromedia.com/coldfusion/7/htmldocs/00000236.htm#3765835>. It offers many additional formatting options that go beyond what was discussed here. You might also want to investigate the subtags to cfdocument: cfdocumentitem and cfdocumentsection. Cfdocumentitem can be used to put page breaks, headers, or footers in your documents. Documentation on the tag is located here <http://livedocs.macromedia.com/coldfusion/7/htmldocs/00000237.htm#3766460>. cfdocumentsection is used in conjunction with cfdocumentitem to give different header, footer, or page numbers to different sections of the finished document. More information on this tag is found here <http://livedocs.macromedia.com/coldfusion/7/htmldocs/00000238.htm#3766659>.

Wrap Up

I recently took over management of the Hartford Connecticut Macromedia User Group. The idea for this usage of cfdocument came out of a demo Ben Forta gave when he spoke at our group a few months ago. Although it doesn't make use of PDF forms, it certainly emulates the functionality available in them, and depending on the quality of your PDF image extracts should be suitable for most applications. 

About the Author

Jeff Houser has been working with computers for over 20 years and in Web development for over 8 years. He owns a consulting company and has authored three separate books on ColdFusion, most recently ColdFusion MX: The Complete Reference (McGraw-Hill Osborne Media).

jeff@instantcoldfusion.com