

Ask-a-Muse: Why Don't My Images Display?

Posted At : December 15, 2006 2:23 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Tips and Techniques

Muse Reader Asks:

"When I have subfolders, I have problems displaying images correctly. I tried setting a var called `#dir_path#` in my application.cfm file (this is actually to differentiate between dev and prod). Then I tried calling my includes and images like so:

```
<cfinclude="#dir_path#/includes/header.cfm"/>

```

I know that with the cfinclude, I can't use a physical path, but even if I use a physical path with my images, it doesn't work - I just get a little box with a red "x" in it where my image should be. I've asked another programmer (who is way better than me) about it, and he said he just doesn't use subfolders because of that problem.

First let me say that you may have overestimated your friends programming skills. There is a fundamental difference between using cfinclude and displaying an image. the difference is that one of them has to do with Coldfusion and the other one simply doesn't have *anything* to do with Coldfusion. Let's review what happens when you load a coldfusion page with an image in it.

Coldfusion Page processing

Let me first warn the purist that we are thinking conceptually here so we are going to bypass that whole "compile-into-classes" bit to avoid confusion. Let's take the following CF page.

```
<cfinclude template="/inc/header.cfm">
  <cfinclude template="bodycode/mybody.cfm">
  
  <cfinclude template="../otherstuff/myfooter.cfm">
```

Here's what Coldfusion does to process this page.

header.cfm using a leading slash

When Coldfusion gets to that first include it sees `/inc`. The first thing it will do is look for a *Coldfusion Mapping* that translates `/inc` into a physical file path like `"c:\myincludes\"` (or `/var/home/myincludes/` for you nix folks). Then it looks for `"header.cfm"` and *includes that file in it's processing*. Notice, it does not simply dump the contents of the file to the buffer. It *executes* the file - it parses the coldfusion in it and makes it a part of the runtime request. If you wanted the contents of the file you could just open it and include it as a variable (see my post on [Cfinclude Vs. Cffile](#) for an explanation of that technique). Whatever is produced by executing the file gets put in the "output buffer". The output buffer is the stuff that gets sent back to the browser. The browser tries to make sense of it and "render" it into something viewable.

If Coldfusion fails to find a mapping called `/inc` it's next step is to look for a mapping called `"/`. This mapping is created by default when you install Coldfusion. Many "shared" hosts remove it because it usually points to the default web root of the

default site. If Coldfusion finds this mapping it will look for a folder called "inc" and use that as the location for the file.

mybody.cfm - a relative path

If you don't use a leading slash Coldfusion understands that you are using the directory of the current tag. Note... *of the current tag*. That doesn't mean the *base template path*. So If I'm 2 or three folders deep and I want something in the root folder of my site, this will *not* give it to me. I will have to use the syntax for backing up.

myfooter.cfm - backing out of the current directory

This syntax allows me to start in the directory of the current template and "back up" *n* number of directories. For example, if the current template is /foo/foo2/foo3/mytemplate.cfm and I need to include /foo/yourtemplate.cfm I could do it with `<cfinclude template="../../yourtemplate.cfm">`.

The Source of Confusion

Now if you are used to working with images on HTML pages and you read this explanation for the first time you would think to yourself immediately, "...that's just like how web servers handle images." That's exactly why it's confusing. There is a similar syntax for images (leading slashes, relative directories and the like). The thing to remember is that images are called when the *browser makes a request for them from the web server*. This request *bypasses Coldfusion Altogether*. So whatever information is needed to get at that image it has to come from the web server.

Let's go back to our example. Say the variable "imagefolder" contained "/images". What happens when Coldfusion encounters the line ``? Does it try to "include" the image somehow? No, of course not. The image is not a part of this request. How does it get on the page you ask? When the browser parses a web page for the first time it takes note of additional resources it needs to render the page like images, JavaScript files and flash. Then the browser makes *additional http requests* for these other resources. A single web page can be responsible for dozens, sometimes hundreds of http requests. In the case of our example CF outputs `` to the browser and the browser sends a request to the web server for /images/mylog.jpg. The web server doesn't use Coldfusion to figure this out at all. It knows the "web root" and it knows of "virtual directories". It looks for an "images" folder off of the web root and finds the file there.

Conclusions

Always think about the web root when working with images and other resources. The "leading slash" means "start at the web root" to the web server, but it means something totally different to Coldfusion. With images you can have leading slashes (absolute path on this domain), full absolute URLs beginning with HTTP (absolute path with domain specified), no leading slash (relative domain... relative directory from which the page is served) and recursive type syntax as in "../" - backing up one directory. this last one can never back up below the web root.

One More Tip

One of the important ways to troubleshoot path problems is to look at the source code. Figuring out what Coldfusion is producing would have shown something like `` obviously that will not work in a web page. If

you think about your web root and folders beneath it you will figure out why your images are not displaying.