

## Using CFINCLUDE

Posted At : February 20, 2005 11:37 AM | Posted By : Mark Kruger

Related Categories: Using Coldfusion Tags

Did you ever get frustrated with the `<cfinclude>` tag? If you started out in HTML you probably have. Why is it that you can put an image in a directory and put a cfm file in the same directory, but not have them work the same? You can call `` and it seems to work just fine, but when you try to do `<cfinclude template="images/blah.cfm">` it doesn't work? Aren't the 2 items in the same path? Nope, and here's why.

When you call `` it does not "include" anything in the page processing. Instead it generates a separate http request AFTER the page is downloaded to the browser. That second request is made by the *browser* using information from the http header. A typical http header might look like this:

```
GET /mycode/myPage.cfm HTTP/1.1
Content-Type: text/html
Referer:
User-Agent: Mozilla/4.0
Host: blog.mxconsulting.com
Content-Length: 488
....
```

The web client (i.e. the browser pardon the pun on "ie") parses out the directory information (/mycode/) and combines it with the host information (blog.mxconsulting.com) and makes a brand spankin' new request for (drum roll please) ***blog.mxconsulting.com/mycode/images/blah.jpg***. This new content comes to the page *after* the original page is rendered and sent to the browser - long after CF has forgotten about the request altogether and moved on to try to run your interminable query for all the books about wax in the library of congress (and I know who you are candle boy). So you see it's the client browser working together with the web server that actually serves up this relative path.





Now let's consider Coldfusion's handling of `<cfinclude template="images/blah.cfm">`. This code works radically different from the `<img>` tag. One might say that it has *nothing to do with* the process whereby images get on your web page (and .js files and .css files and your flash intro movie that we all hate, but we tolerate it because we love you... and just because you think that 20 minutes of the halleluiah chorus in 4 bit sample on a honky tonk tinny piano is a good idea we are willing to give you a break... but I digress).

As you may or may not know, CF parses through files with a .cfm extension by default - and yes I know you can parse any extension if you are part of the secret cadre of web masters and professionals who care passionately about making everyone believe your pages are something other than what they are - so lay off on the hate mail. When CF encounters a tag that begins with `<CF` it executes that tag as a command. When CF comes across a `<cfinclude>` tag it *parses that file and runs whatever code it finds in it*. When does this happen? It happens **before** IIS gets any content back from CF (minus the use of cfflush - but that's a different blog). Remember, CF executes on the server and it is autonomous from the web server in a way.

The web server in the process, doing its job, sees a .cfm file extension and says, "hey, that's not my job, that belongs to Bob, the CF server engine.... hey bob, come over her and check out this file - it looks like one of yours." Bob, not wanting to be a party pooper, and liking his reputation as a stand-up guy says "...Your right sam, this is one of mine. Let me have it and I'll get back to you in a few milliseconds ok?". Sam agrees, because he works for Microsoft and he's really only interested in playing monopoly so any work he can hand off is fine with him. Bob takes the request and parses through it looking for `<CF` tags to execute - which he does. `<cfinclude>` are executed *inline* with other code - exactly like they were one long template - and they are. They share the same scope and variables and even have (on occasion) a toothbrush.

Bob isn't just executing, he's building an output buffer. That buffer is one long string that is going to serve as the "content" of the http result that gets sent back to the header. When the entire page is executed, Bob taps Sam on the shoulder and says, "Hey Sam, I have the results back on that .cfm page you gave me - here you go." Sam, who was on the verge of landing on Boardwalk and being trounced by Apache says "Ok, I needed a break, I'll send that off right away." And off goes the HTML to the client browser.

How does Bob know which file to include? There are 2 answers and they follow an order of precedence. First, Bob looks in the CF settings to see if there is a *mapping*. Mappings are set in the CF administrator and they create a *logical path* that is pointed to a *physical* path. The interface looks like this:

Active ColdFusion Mappings		
Actions	Logical Path	Directory Path
 	/	E:\inetpub\wwwdocs\
	/CFIDE	E:\inetpub\wwwdocs\CFIDE
 	/FK_ClassLibrary	E:\RegencySites\fk_live\Clas

In this example the mapping /cfide mapped to a physical directory. If I did an include like `<cfinclude template="/cfide/blah.cfm">`, it would look in the physical directory that *means* /cfide to the cf server for the file "blah.cfm". By default, CF maps the *Default web site* to the slash. If you are on a server where your site IS the default this is fine - in fact it causes the `<cfinclude>` tag to work much like the `<img>` tag.

If no mapping is found in the settings, CF starts with the *current* template path and uses relative path markings to find the file. So, images/blah.cfm called from the /mycode/ folder would look for /mycode/images/blah.cfm. If you were calling a file from the root folder of your web site and YOU are not the default web site mapping for that server, you cannot use /blah.cfm (as you might think). Instead, if you are in mycode, you must use `../blah.cfm`.