

Portable ColdFusion Code (Linux and Windows)

Posted At : June 21, 2008 12:35 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Tips and Techniques

I'm a big fan of ColdFusion on Linux. Not that I know half as much about Linux as I do about Windows. Still, we have a good number of Linux servers here at CF Webtools and CF Linux Guru [Ryan Stille](#) keeps them all humming a happy tune. I think both platforms have advantages. If you have ever tried to write ColdFusion code that is able to run on *both* Linux and Windows you will know there are a few differences. One difference is case sensitivity for file names.

On windows if you include a file called "myfile.cfm with a cfinclude that is something like `<cfinclude template="myFile.cfm">` it will work just fine. Move the same code to Linux and it will generate a "File Not Found" error because of the capital "F" in your include statement. The good news is that once you fix this problem on Linux you can move it back to Windows and it will now work on both platforms. The other common problem is a bit more challenging. It has to do with file paths.

The Path Conundrum

On Windows, file paths usually look like you are accustomed to seeing them on your own windows desktop (Mac and Linux desktop users please sit on your hands - I'm not talking to you right now). The location of your files is usually something like "C:\mywebsite\mydomain\www" - or whatever convention you have chosen. Sometimes files are located elsewhere on the network and there is a UNC path, but I would say that the majority of file operations happen on direct storage with a drive letter. Windows servers typically have only 1 or 2 partitions (a "c:" drive and a "d:" drive for example). Linux servers are quite different.

In Linux there are multiple partitions. Each one starts with a mount point like "/opt" or "/etc" (Linux users please forgive me if I get any of this wrong and be kind with your comments). So the root of your web site on Linux will probably look like "/opt/home/WWW-data/mydomain/www". You will note that 2 things are missing. First, there is no drive letter that is roughly analogous to a partition or a physical disk. Instead there is "/opt" - which looks like a directory except that it comes first. This can be confusing if you come from the windows world because it doesn't seem like a real starting point. Second, you will notice that the slashes are front slashes and not back slashes.

So if your windows code looks something like this:

```
<cfset path = "c:\mysites\mydomain\www\images\products"/>

<cffile action="READ" file="#path#\sk1001.jpg" variable="x"/>
```

Linux is going to error out with a file not found error because there is no "c:\" starting point. Ok - so we could solve the "starting point" problem with expand path. Let's give that a try.

```
<cfset path = expandpath("../") & 'images\products'/>

<cffile action="READ" file="#path#\sk1001.jpg" variable="x"/>
```

That gets us half way there doesn't it. We now have a path that looks like

/opt/home/WWW-data/mydomain/www\images\products

Unfortunately Linux will still choke on it because of the backslashes. To make it work we will have to change it to:

```
<cfset path = expandpath("../") & 'images/products' />
<cffile action="READ" file="#path#/sk1001.jpg" variable="x" />
```

(Notice the front slashes)

Windows Doesn't Care

Here's what you may not know - especially if you have only ever moved code from Windows to Linux and not the other way around. Windows not only doesn't care about case, it doesn't care about slashes either - at least not from within your CF code. On windows the code above produces the following file path:

C:\mywebsite\mydomain\www\images/products/sk1001.jpg

This actually works fine. Windows (or perhaps the JVM or ColdFusion engine is translating) sees either the forward or the backslash as valid path delimiters. This saves a lot of headaches. I have seen code written with a path delimiter as a variable just to compensate for the slashes - as in:

```
<!-- in application.cfm/cfc --->
<cfset application.pathdelim = "/">
<!-- in cfm files --->
<cfset path = "#dir1##application.pathdelim##dir2#" />
```

An E for effort to be sure.

Rules of thumb

Here are some of the Muse rules of thumb for writing code that works on both platforms.

- If possible, write your code on the Linux platform and port it to Windows. Linux being more picky and Windows more forgiving, this will result in fewer debugging issues.
- Pay attention to case issues for cfinclude, cfmodule, cffile, css, images, swf files etc. Pretty much anything that points to a "file" should account for case.
- Use ExpandPath() to correctly variablize directories.
- If ExpandPath() is not an option, use variables set in the application.cfm/cfc file (or an ini or xml file) for file locations. This gives you one spot to alter when porting your code.
- Use the front slashes for file operations - it will work on both platforms.

Of course there are other issues that will be important but these 2 are the most common *coding* issues that arise. Now I'm sure some of the Linux cabal will want to comment. I bought tar and feathers on rebate this morning at Menards just to save you the trouble. So let fly the arrows of discontent and I'll just sit here huddled under my shield of pragmatism :)