

CFHTTP and DNS Revisited

Posted At : January 28, 2009 7:10 PM | Posted By : Mark Kruger

Related Categories: ColdFusion

The muse has a few posts that are more popular than others. The post on [Update Using a Join](#) is legendary (in my own mind) and has achieved popularity even on SQL forums and blogs. My posts on interfacing with a HP e3000 or using Webspere's MQ Series.... eh... not so much. Among those posts on the *popular* side of the spectrum is the post on [Troubleshooting CFHTTP](#). This little gem is hit more often than a quarterback for the Detroit Lions. Why? Because CFHTTP relies on a number of things that are outside of ColdFusion's control - like DNS, Networking and sometimes SSL. Today I was on the phone helping a site owner come to grips with just such a problem.

The admin was working with me to figure out the following. They had a script that used CFHTTP to create a request to a URL that resided on the same web server. The Apache server was configured with virtual host headers and the server supported multiple sites - let's say the sites were abc.com and def.com. When the user was on abc.com, the CFHTTP call hit a script on it's own domain - abc.com. If the user was on def.com the server used that domain in it's CFHTTP call and so on.

```
On abc.com
<cfhttp url="http://abc.com/myscript.cfm"/>

on def.com
<cfhttp url="http://def.com/myscript.cfm"/>
```

Now the script being pointed to (myscript.cfm in the example above) was the exact same file in both cases. The only difference was the domain. When abc.com was hit the script returned instantly. When def.com was hit the script hung and timed out with a "connection failure". What was wrong? If you are ahead of me you will have already made a guess. I'm betting you guessed "DNS issue" - just like I did.

The crew working on the issue insisted that they had checked resolution on the CF server and it was able to resolve the domains in question. In fact, this same code was working correctly in a previous environment. "Hmmm....", I said in my best I'm-a-high-paid-consultant-and-I-know-better-than-you voice. "Just for me, let's go through it again." I had the admin SSH into the web server (a Linux box) and use WGET to retrieve the exact file specified in the URL that was causing us trouble. Unfortunately (for me) this work splendidly. The request brought back the content and it was obvious to the whole crew that I was as clueless as they were.

In the midst of our continued head scratching, I went back to the test script and tried again. To my amazement, although it had been timing out not 3 minutes earlier, this time it returned instantly with the information needed. What happened? What had changed? Well, one of the folks on the crew thought it might be a good idea to see some Java debugging. They had added a switch to the JVM.config file and... (you guessed it) restarted the ColdFusion server. More information revealed that they had changed the DNS settings in an attempt to resolve the issue the day before, with no success. What they did not guess is that, once a CFHTTP call is attempted, Java caches the resolved IP address and reuses it. In this case, even though our tests on the server showed the server was able to resolve the domain, the JVM still had hold of an old IP address and was attempting to use it. A restart reloaded the server and cleared the cache fixing our problem and leaving me with a egg facial.

So my new rule of thumb - whenever making DNS changes that affect CFHTTP, CFFTP, CFMAIL etc. Don't forget that you may need to restart CF. Now perhaps one of my Java savvy readers would like to post a snippet of code that shows how the cache can be purged without a restart. Anyone?