

Trusted Cache and Debugging

Posted At : July 18, 2007 2:21 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Troubleshooting

Hello Muse readers. It's been a long time between posts and that's quite unlike me. I've been vacationing, traveling, and swamped with new business. My biggest headache at the moment is finding advanced developers to join my team so if you know anyone send them my name. We are a Coldfusion shop doing high level Coldfusion and Flex work and we need 2 advanced developers as we speak. We have great wages and benefits and Omaha is one of the top 10 places in the country to live - just ask my friend and neighbor Warren Buffet (ha). Meanwhile, here's today's tip.

Let's suppose you have a production CFMX server where trusted cache is turned on. At some point you have to do some quick troubleshooting due to an error so you also turn on debugging. In the debug information you see the error detail and you know exactly what to do to get it fixed. What's your next move? Of course, using CFEclipse or Homesite or (gulp) Dreamweaver you open the file and make a change - then you refresh the page. What happens? Actually 2 very contradictory things happen.

The Fix is *Not* In

Even though you made a change the error is still generated because you have failed to turn off the trusted cache. That's how trusted cache works. Without trusted cache turned on, CF checks the modified date of the file and if it has changed it recompiles it. With trusted cache turned on however, the file is loaded once and the date is never checked. This allows trusted cache to reduce the overhead of the scripting engine. On a busy server (especially one with a busy disk system) this can be a measurable benefit.

Debug Information *May* Show the Fix

This is the confusing part. Even though the error has been generated you may see the changes you made in the debug information - especially if you have the "robust" information turned on. Remember your files are compiled into Java byte code. Some meta data is also included in that byte code that allows the debugger to identify the line number (sort of) of the original file where the error occurred. But the original file itself is not included - so the debugger actually opens the original file and teases out the 4 to 6 lines of code around the error to display. Since the file has changed you see your changes in the debug information. The result is that you scratch your head and say - "Hmmm... obviously the server sees my changes - why didn't my fix work?"

If you are still confused try this. Create a file with an error in it.

```
<cfparam name="url.userid" default="" />
<!--- single quote throws and error --->
<cfquery name="test" datasource="test">
    SELECT      *
    FROM        users
    WHERE       user_id = #url.userid#
</cfquery>
```

Turn on the trusted cache and then run the template. At this point your Java code (along with the error) is cached in the trusted cache. NOTE: it has to be an error that will not prevent the template from compiling. The error above is only thrown if you

don't pass in an int. You may actually have to get this file to run successfully (compile into bytecode in other words) before you throw errors. So you could pass it `userid=1` and then `userid=x` to throw the error. Notice in the debug error code that you can actually see your query - usually with line numbers to the left of it. The `user_id = #url.userid#` is clearly visible.

Now open the file and "fix" the error. Something like this will do it:

```
<cfparam name="url.userid" default="" />
<!-- single quote throws and error -->
<cfquery name="test" datasource="test">
    SELECT      *
    FROM        users
    WHERE       user_id = #val(url.userid)#
</cfquery>
```

Try running the file with an incorrect `url.userid` variable (pass in `userid=x` or some other string). You will find that the error is generated exactly as before even though you fixed it, but the debug information will clearly show your fix (`user=#val(url.userid)#`). If you did not think of the trusted cache you might run off looking for some other reason as to why the page is erroring out.

The main lesson is, when you enabled debugging on a server check the trusted cache.