

Datasource Attribute in Application.cfc

Posted At : May 9, 2012 2:19 PM | Posted By : Mark Kruger

Related Categories: ColdFusion

You may know about the "datasource" property in ColdFusion 9. It allows you to create a variable in your Application.cfc file called "datasource" and then skip the "datasource" attribute of your query tags. That's pretty neat. Here's how it works. In the Application.cfc properties...

```
<!---Application.cfc-->
<cfscript>
    this.name = 'mort';
    this.sessionmanagement = true;
    this.datasource = 'myDsn';
</cfscript>
....
```

Then in any component that is "inside" of my application "mort" above I can do something like this:

```
<!--- myMethods.cfc --->
<cffunction name="getAllUsers">
    <cfset var 'myQry = ''/>
    <cfquery name="myQry">
        SELECT username, email
        FROM      users
    </cfquery>
    <cfreturn myQry/>
</cffunction>
```

Do you notice what is missing? There's no "datasource" attribute in the cfquery tag. ColdFusion automatically picks up the datasource from the Application.cfc instead. It's a nice time saving effort that reduces code and allows for fewer mistakes. Good for multi-tenant code too.

A Minor Detail

There is one issue that I was made aware of with this approach. Super genius guru Phillip Senn had a head scratching problem where he would set this variable in Application.cfc. Periodically he began getting an error which said:

```
The value of the attribute datasource, which is currently "" is invalid.
```

He tried some different things - working with the application names and extended properties of the Application.cfc but nothing seemed to help. Finally he ran the **var scope checker** which identifies places where unvarred variables exist inside of CFCs. He found several unvarred variables. After he fixed them the problem went away.

So if you are using this Application based *this.datasource* approach and you get some random errors where the variable seems undefined - or seems defined as a blank string - start looking for vars that are not properly scoped within your components. It may just fix you right up.