

ColdFusion scoping issues and the Var scope

Posted At : 28 October 2009 15:51 | Posted By : Shaun McCran
Related Categories: Development, Coldfusion, Best practices

One of the more common problems with a ColdFusion application, especially when it starts to grow or involves more than one developer is variable scoping, and scope bleed through issues. More recently I have had to re examine the framework of an application, in an effort to track and eliminate any instances of this.

Not properly scoping a variable, IE:

```
<cfset myUnscopedVar = "Foo">
```

Forces ColdFusion to add the variable to the 'variables' scope. This scope is available to the entire template. This is even more important when you consider CFC's and the potential damage that a leaked variable could cause. One of the major benefits of OO in ColdFusion is the encapsulation that it provides. As soon as you start declaring un scoped variables, then you lose that.

Mike Schierberl has quite an in-depth blog about this subject, <http://www.schierberl.com> , he goes into a great deal of depth about this subject, and he has built and maintains the varScoper project. The varScoper allows you to scan your code base for un scoped variables, and after a quick inspection of its output it seems very reliable. I'm not sure its 100% foolproof, but would be a valuable addition to a QA process.

Changes to the Var scope in Coldfusion 9

I am currently not using ColdFusion 9 in any environment; in fact my hosting company is still on 7! That said it appears that there are some significant changes to the Var scope in ColdFusion 9.

Ben Forta sums it up nicely here:

<http://forta.com/blog/index.cfm/2009/6/21/The-New-ColdFusion-LOCAL-Scope>

He does touch on an interesting point in the article above. I quite like the structured layout of a CFC, and part of that regime for me is declaring all the Vars at the top of a function. It sort of matches the layout of an Action Script class as well, declaring all your constants etc at the top of the script. It is also a very easy coding standard to adopt and enforce if you are part of a larger team.