Build A Better ColdFusion App: When The Var Doesn't Exist

Posted At : October 8, 2013 12:27 AM | Posted By : Cutter
Related Categories: Build A Better ColdFusion App, ColdFusion

***Author's Note**: Due to some of the feedback below, I'm going to remove a bit of example regarding paraming objects (see the feedback). Some have actually benchmarked where this creates overhead, rather than improvement of performance, so I'm going to stick with simple variables as this is researched further.*

So, let's kick off this round of "Build a Better CF App" with a simple bit of code that you might see quite often in an app:

```
<cfif NOT StructKeyExists(FORM, "isactive")>
 <cfset FORM.isactive = false>
</cfif>
```

Simple, right? Probably dozens of these hanging around in your code. Now, let's look at a simpler form, that also gives us the added benefit of type checking.

```
<cfscript>
 param name="FORM.isactive" type="boolean" default=false;
</cfscript>
```

What? Huh? Yeah, this makes sense. See, a checkbox type formfield doesn't actually get passed on form submission if it isn't checked, so chances are you're running this same code, in one way or the other, to default that value for you. Utilizing the *param* here (*cfparam* in tag form, but who's using tags outside of display stuff anymore? [hint, hint]) is really the better way of handling this.

1. You get a default value for the variable
2. The default is overridden, if the variable actually occurs
3. If the variable does exist, the *param* ensures that it is of the proper type

But wait, let's throw in another similar bit of code for good measure. How many times have you seen something like this?:

```
<cffunction name="getUser" output="false" returntype="User">
 <cfargument name="firstName" type="string" required="true">
 <cfargument name="lastName" type="string" required="true">

 <cfif NOT StructKeyExists(ARGUMENTS, "userID")>
  <cfset ARGUMENTS.userID = 0>
 </cfif>

 <!--- More code --->
 <cfreturn LOCAL.User>
</cffunction>
```

Just doesn't feel right, does it? Especially when ColdFusion already gives us a way to handle this scenario:

```
public com.mysite.User function getUser (required string firstName, required string lastName, numeric userID=0) {
 // More code
 return LOCAL.User;
}
```

You can apply a default value to any *argument* of a *function*, utilizing your arguments just as you would a *param*.

Now, there may be some valid reason for *not* paraming a variable, and using conditionals in some way similar to those above. They're few and far between, but there are exceptions to most every rule, so you'll need to evaluate that when the time comes.