

Why IIF() Should be Avoided

Posted At : May 17, 2010 11:37 AM | Posted By : Jon Hartmann

Related Categories: ColdFusion

I was recently reviewing some code from one of my client projects and saw that the previous developer had liked to use IIF() functions to handle the conditional switching of things like button labels. I idly quipped on [Twitter](#) that Every time you execute IIF() an angel loses its wings. A few minutes later a request for some explanation came in on Facebook: Why's that? I like IIF(). Here is why you should avoid IIF().

Its Evaluate()

According to the [Adobe ColdFusion 8 Documentation](#), IIF() is equivalent to the following code:

```
<cfif condition>
<cfset result = Evaluate(string_expression1)>
<cfelse>
<cfset result = Evaluate(string_expression2)>
</cfif>
```

What that means is that each time you execute IIF, its calling Evaluate(). I've read a few places, such as [Ray Camden's Blog](#), that the performance of Evaluate isn't as **bad as it used to be**, but it still can't be good; dynamic execution of code costs processing power because it has to be computed at run-time.

Its Evaluate(), part 2

Under the hood, IIF() is executing an Evaluate() against at least one of your parameters. If you don't have to run Evaluate() to solve your problem, you don't need to call IIF() to do it either, and **the places where you have to use Evaluate() are few**. In fact, I'd bet that almost all uses of IIF() are to conditionally display strings, something like this:

```
<cfoutput>#IIF((Query.RecordCount gt 1), DE("Records"), DE("Record"))#</cfoutput>
```

See what you have to do there? You have to escape your strings just to use IIF(). If you wouldn't need Evaluate() to handle the logic of your code, then you don't need IIF().

Easy String Switching

If you're using IIF() to handle string switching like the example above because it's shorter to type, you're better off writing something like **this utility function I wrote**. Its going to save you some processing power, and it's actually shorter:

```
IIF((Query.RecordCount gt 1), DE("Records"), DE("Record"))
IfElse((Query.RecordCount gt 1), "Records", "Record")
```

Summary

1. Using IIF() is probably slower than a regular if/else block because its using Evaluate().
2. IIF() uses Evaluate() so you shouldn't use it unless you need to Evaluate().
3. IIF() actually takes more typing than similar "inline" case solutions.

So, just remember: Every time you execute IIF() an angel loses its wings.

I also found **quote from Sean Corfield about IIF()** that makes it sound like IIF() actually Evaluate()s both the 2nd and 3rd argument both. I seriously hope he was mistaken or this quote is too old to be applicable.