Tips on CF Script

Posted At : March 24, 2008 1:00 PM | Posted By : Mark Kruger
Related Categories: Coldfusion Tips and Techniques

From a reference on an email list I stumbled onto Pete Freitag's **cheat sheet** for CFSCRIPT and I thought it was worth mentioning. I use Cfscript for any largish block of logic. With the advent of CFC's it has become even more useful. I think the inline commenting is more readable and the syntax translates to other languages with a few modifications. To Pete's tips I would add the following:

Looping through a database query - cfloop or cfoutput notwithstanding, sometimes it's nice to do this in CFSCRIPT.

```
<cfscript>

    for(rec = 1;
        rec LE qry.recordcount;
        rec = rec + 1) {
    // handle a column
    thiscol = qry['*columname*'][rec];

    }
</cfscript>
```

Using a Component.

```
<Cfscript>
    // instantiate only
    myObj = createObject("component","com.dosomething");
    //instantiate and call a function
    myObj2 = createObject("component","com.dosomething").init(args);
    //use something from the object
    strMsg = myObj.getMsg();
</cfscript>
```

Notice the idea of calling a function in-line with the create. This is pretty ordinary stuff for Java folks, but it sometimes escapes Coldfusion developers who do not typically use CFSCRIPT. But using a script block makes these conventions translate pretty well between languages.

I also like CFSCRIPT for blocks of variable "set" statements. This is especially useful at the head of a function. Instead of:

```
<cffunction ....>
    <cfargument ...>
    <cfargument ...>
    <!--- set up localized vars --->
    <Cfset var retStr = structNew()/>
    <cfset var qryCases = ''/>
    <cfset var qryImport = ''/>
    <cfset var qryTot = ''/>
    .... etc
```

You can do the followwing:

```
<cffunction ....>
    <cfargument ...>
    <cfargument ...>
    <Cfscript>
```

```
        //dim local variables
        // return var
        var retStr = structNew();
        // Queries needed
        var qryCases = '';
        var qryImport = '';
        var qryTot = '';
        var qryRid = '';
        //other vars
        var importCases = '';
        var args = structNew();
        var col = '';
        var case_id = '';
        var import_id = addNewImportRec(type="Access");
        //defaults
        ret.err = 0;
        ret.msg = '';
    </CFSCRIPT>
```

Notice that you don't have to break out of your code block to begin using the vars you have just created. The only rule is that the "var" type statements (the var keyword makes the variable a part of the function scope an inaccessible outside the function). This makes for less typing and is more readable (at least I find it so).

One more note. Like other languages CFSCRIPT does not require the curly braces ("{}") if you have a single line operation. This allows for neater "IF" statements as in:

```
<cfscript>
    // select bob's car
    if(someVar IS 'makeSad')
        bobsCar = 'pinto';
    else if (someVar IS 'makeHappy')
        bobsCar = 'BMW';
    // set up bob's demeanor.
    if(bobsCar IS 'pinto')
        bobsDemeanor = 'sad';
    else if(bobsCar IS 'BMW')
        bobsDemeanor = 'Happy';
</cfscript>
```

Notice that because I'm only doing something that requires 1 line I do not need the curly braces anywhere. I use that extra line for comments.