

tag.cfc 0.1: Write Your Own Code Generator

Posted At : December 12, 2005 2:22 PM | Posted By : Steve

Related Categories: ColdFusion

Back when I had some free time, I started working on my own code generator. Partly because other code generators create code slightly different from my preferences and partly for the challenge. I have since gotten too busy to finish it, but I went ahead and finished the kernel component of the system - the one that actually generates the code.

One advantage of ColdFusion being a tag-based language is that generating code for ColdFusion largely means writing tags.

The best way that I can think to show how tag.cfc works is by example. `cf_sebForm.cfc` and `cf_sebField.cfc` are both components that are inherited from tag.cfc. They are each used to generate the code for my [cf_sebForm](#) and [cf_sebField](#) custom tags.

These components are actually very simple. They extend tag.cfc and each have two methods: `vtml()` and `schema()` which return the VTML for the tag and the XML Schema for the tag, respectively.

Here is some example code using these components to output the code needed for these custom tags.

```
<cfset sebForm = CreateObject("component","cf_sebForm").init()>
<cfset fields = ArrayNew(1)>

<cfset sebForm.setAttribute("formname","myform")>
<cfset sebForm.setAttribute("librarypath","/lib/")>

<cfset fields[1] = CreateObject("component","cf_sebField").init()>
<cfset fields[1].setAttribute("fieldname","blah")>
<cfset fields[1].setAttribute("label","Hello")>
<cfset fields[1].setAttribute("type","text")>
<cfset fields[2] = CreateObject("component","cf_sebField").init()>
<cfset fields[2].setAttribute("label","Submit")>
<cfset fields[2].setAttribute("type","submit")>

<cfset sebForm.addTag(fields[1])>
<cfset sebForm.addTag(fields[2])>

<cfoutput>
<pre>#HTMLFormat(sebForm.write())#</pre>
</cfoutput>
```

The code that this code would output follows:

```
<cf_sebForm formname="myform" librarypath="/lib/">
  <cf_sebField fieldname="blah" type="text" label="Hello" />
  <cf_sebField type="submit" label="Submit" />
</cf_sebForm>
```

In order for the previous code to be useful, of course, you wouldn't want the values to be hard-coded. You would need to get those values from a form or from data in a database, for example. While tag.cfc doesn't handle that part for you, it will handle much of the last-step of code generation once you have gathered that information.

Incidentally, when I was working on my own code generator (a task I hope to return to someday), I used my [DataMgr component](#) to get the structure of the database. Since it works the same across multiple databases, it makes for a nice cross-database solution.

You can [download tag.cfc](#) from my site. I also have two sets of tag CFCs that inherit from tag.cfc. The

first, "CFCs" includes components to generate cfcomponent, cffunction and cfargument. The second, "sebtags" is a set of CFCs used to generate my [custom tags](#).

This code illustrates my best understanding of good OO code, so it should be a good example of such. That being said, if someone can show why it isn't a good example of such, let me know.

Good luck!