

Global Custom Tag Attributes

Posted At : December 5, 2007 8:00 AM | Posted By : Steve

Related Categories: ColdFusion, sebtags

While working on my custom tag set, I discovered that I often have attributes that will be the same across a whole site. These attributes need to change from site to site.

For example, I have a "skin" attribute to skin the output of my custom tags. Every tag in a site should use the same skin. If I copy some code from one site to another, it should automatically use the skin of the site to which it is copied.

I started by setting attributes using variables:

```
<cf_sebForm skin="#request.skin#">
...
</cf_sebForm>
```

This worked well enough, but I ended up with code that I felt was cluttered. I didn't want to see the skin attribute on every single form. It was taking visual space from information that was important about that form.

So, I decided to create a special structure to set default attributes. It was important that I set a default attribute externally instead of overriding an attribute set in the tag itself. The closest code should still win. This also allows me to override the global attribute default for a particular instance if needed.

All of my custom tags look for a "cftags" structure in request scope (something that I hope could become a standard). Within that, my tags look for a structure called "sebtags".

So, the following code would effectively set a default value of "plain" for the skin attribute of any of my custom tags:

```
<cfset request.cftags = StructNew()>
<cfset request.cftags.sebtags = StructNew()>
<cfset request.cftags.sebtags.skin = "silver">
```

Each individual tag has its own structure within cftags as well. For example, if I wanted to set the skin for only my **cf_sebForm** custom tag, I could use the following code.

```
<cfset request.cftags = StructNew()>
<cfset request.cftags.cf_sebForm = StructNew()>
<cfset request.cftags.cf_sebForm.skin = "silver">
```

If both sebtags and a structure of the tag name itself are used, then the structure for the particular tag ("cf_sebForm" in this example) takes precedence. Local attributes still override both, however.

Here is some modified code from one of my tags demonstrating how I accomplish this:

```
<cfset TagName = "cf_sebTable">
<cfif ThisTag.ExecutionMode eq "Start">
    <cfif StructKeyExists(request, "cftags")>
        <cfif StructKeyExists(request.cftags, TagName)>
            <cfset StructAppend(attributes, request.cftags[TagName], "no")>
        </cfif>
        <cfif StructKeyExists(request.cftags, "sebtags")>
            <cfset StructAppend(attributes, request.cftags["sebtags"], "no")>
        </cfif>
    </cfif>
</cfif>
```

```
<cfparam name="attributes.skin" default="">
<!-- more cfparams here -->
...
</cfif>
```

Here is how it works:

The attributes structure always exists in a custom tag. I start by checking for `request.cftags[TagName]`. If that exists, I append the request structure with that structure. It is very important that the third argument "no", otherwise any attributes set when calling the tag would be superseded by the request scope. This is repeated for `request.cftags.sebtags`. Again I use "no" for the third argument so that `request.cftags[TagName]` has priority.

I do all of this before any `cfparam`. This is so that the `cfparam` values will only be used if no value is provided in the tag or in either of the request structures used to set defaults (as those have to take priority over the internal defaults or they would be useless).

I have been using this approach for a few years now and it has saved me from plenty of redundant code.

Next time that you are writing custom tags that you use often and for which you repeat the same attribute several times, give this approach a shot. It could really make your life easier.