

Utility Function: StructMerge

Posted At : May 5, 2012 9:03 AM | Posted By : Jon Hartmann

Related Categories: ColdFusion, Utility Function

If all you need to do is merge two simple structs, then StructAppend() is a great function to get you there, but if you have complex structs (meaning structs that contain structs) you may be wanting to find a solution that does a deep merge:

```
<cffunction name="StructMerge" access="public" output="false" returntype="struct">
    <cfargument name="Struct1" type="struct" required="true" />
    <cfargument name="Struct2" type="struct" required="true" />
    <cfargument name="Overwrite" type="boolean" required="false" default="true" />

    <!-- Loop Keys -->
    <cfloop collection="#Arguments.Struct2#" item="Local.Key">
        <!-- Find if the new key from Struct2 Exists in Struct1 -->
        <cfif StructKeyExists(Arguments.Struct1, Local.Key)>
            <!-- If they are both structs, we need to merge those structs, too -->
            <cfif IsStruct(Arguments.Struct1[Local.Key]) AND IsStruct(Arguments.Struct2[Local.Key])>
                <!-- Recursively call StructMerge to merge those structs -->
                <cfset StructMerge(Arguments.Struct1[Local.Key], Arguments.Struct2[Local.Key], Arguments.Overwrite) />
                <!-- We already checked that the key existed, now we just check if we can overwrite it -->
                <cfelseif Arguments.Overwrite>
                    <cfset Arguments.Struct1[Local.Key] = Arguments.Struct2[Local.Key] />
                <!-- The unused case here is if Overwrite is false, in which case Struct1 is not changed -->
            </cfif>
        <!-- If it doesn't exist, you're free to merge -->
        <cfelse>
            <cfset Arguments.Struct1[Local.Key] = Arguments.Struct2[Local.Key] />
        </cfif>
    </cfloop>

    <cfreturn Arguments.Struct1 />
</cffunction>
```

This code merges all sub structures as well as the core one. I'd also say it would be an easy adjustment to merge arrays and other data types if necessary. I'm not sure if CF10 will be adding something like this or not, but I know for CF9.0.1 I needed it.