

# Dynamically Adding Methods to a CFC

Posted At : December 7, 2009 1:15 PM | Posted By : Steve

Related Categories: ColdFusion

When ColdFusion 9 came out, I found that [FileMgr.cfc](#) would not work on it. For one function, I had directly copied the [DirectoryList UDF](#) by [Ray Camden](#). This allowed me to use cfdirectory recursively in CFMX (the "recurse" attribute not being available natively until ColdFusion 7).

This worked great until ColdFusion 9 came out. ColdFusion 9 added a built in function named (you guessed it!), DirectoryList. As a result, the FileMgr.cfc threw an error whenever it was instantiated on the code attempting to create a DirectoryList method. The problem here is that I have a lot of external code using that method, so renaming it would be a bit difficult. I would have to track down and change code in several different sites on several different servers. Not only that, but as FileMgr.cfc is publicly available code, everyone using it would have to do the same. Needless to say, this isn't a desirable outcome.

Fortunately, ColdFusion makes it easy for me to maintain my API and still avoid the error telling me that I can't have a function named "DirectoryList". The surprising thing about this is how easy it is.

The first thing that I do is rename my DirectoryList method:

From:

```
<cffunction name="DirectoryList" ...>
```

To:

```
<cffunction name="getDirectoryList" ...>
```

I already have an "init" method in my component that is called any time the component is instantiation. I simply add the following line to it:

```
<cfset This["DirectoryList"] = getDirectoryList>
```

This works because a function in ColdFusion is actually stored in a variable with the same name as the function. You can actually even pass a function around as arguments to other functions - really neat stuff. Now, when you dump the instantiated FileMgr component it will show the DirectoryList method as being available. Calls to that method will work as well.

The only drawback here is that the DirectoryList method will not show up in a getMetaData() call on the instantiated component. Even so, I consider that acceptable as the component will still work correctly.

The above change, however, will not help for internal calls to the "DirectoryList" method. To do that, either change the method calls from "DirectoryList" to "getDirectoryList" or add the following line to the init method:

```
<cfset Variables["DirectoryList"] = getDirectoryList>
```

This technique can also be used to inject a method into a component from external code, but that is a story for another day.