

## Scopes in a CFC

Scope	Purpose/How it Works/Etc
Variables	The variables scope is available to the entire CFC. A value set in the variables scope in one method, or in the constructor, will be available to any other method, or the constructor, in the CFC. I typically use variables in much the same way I use Application variables in an site.
This	The This scope acts like the Variables scope in that it is "global" to the CFC. However, it is also accessible <i>outside</i> the CFC. So if foo is an instance of the CFC, and you do: foo.name = "Rabid" outside of the CFC, then you have just creating a key called "name" in the This scope with the value of "Rabid." You can also cfoutput the value of foo.name. Because of this accessibility, many folks recommend against using the This scope, and instead suggest relying on methods to set data (or get data) inside the CFC. These methods then write to the Variables scope. To repeat: <cfset foo.name = "Rabid"> (where foo is an <b>instance</b> of the CFC) is the same as <cfset this.name = "Rabid"> <b>inside</b> the CFC.
Var	Var scoped variables exist only for the duration of the method. <b>You must use the var scope for any variable that should exist only inside the method, like query names, loop iterators, etc.</b> Sorry to "shout" but the lack of var scoping is one of the trickiest things to debug when things go wrong. Unlike other scope, you do not prefix the scope name in front of the variable.
Arguments	The arguments scope consists of every argument passed to the method. So if the calling code did foo(name="King Camden"), then the foo method will have a variable called arguments.name.
Form, URL, Application, Session, Server, CGI, Client, Request, Cookie	These scopes act exactly as they do anywhere else. <i>In general</i> you should not use these scopes inside a CFC. When you do you are making your CFCs less portable between applications.
Caller, Attributes	Caller and Attributes do not exist inside a CFC. They should only be used inside custom tags.
Super	While not a variable scope, the super scope is a pointer to the methods in the CFC that acts as the parent to the current CFC. So if a CFC inherits a CFC with similar methods names, the child CFC can refer to the parent's method by using super. Example: <cfset result = super.somemethod(>