

# Modularity Shoot-Out

CFinclude vs CFmodule vs CFfunction vs  
Custom Tags

Nolan Erck  
Southofshasta.com

NCDevCon, September 2011

# Overview

- Look at the pros and cons of each tag.
- Situations when one tag makes more sense than the others.
- Differences in memory management.
- Differences in speed (kind of).

# Who Am I?

- Independent Software Consultant ([southofshasta.com](http://southofshasta.com)).
- Sacramento CFUG Co-Manager.
- 15 years of software development.
- ColdFusion since version 4.5.
- 5 years of C++ in the video game industry.
- Past projects: [Riffage.com](http://Riffage.com), [iConvention](http://iConvention.com), [Grim Fandango](http://GrimFandango.com), [SimPark](http://SimPark.com), [SimSafari](http://SimSafari.com), etc.
- Working musician.

# Why am I doing this?

- Work on a variety of projects for different clients.
- Still lots of procedural code in Production!
- Not everyone is writing OO code (yet).
- Let's help everyone improve their code!

# cfinclude

- Pros
  - Really easy to use.
  - Fast.
  - Compatible with pretty much every version of CF.
- Cons
  - Doesn't have it's own scope.
  - No obvious indication what a tag needs (or changes!)
  - Dangerous when dealing with variables.

# cfmodule

- Pros
  - Like a “safer” version of cfinclude.
  - Can pass variables in as attributes.
  - Has its own scope, variables are safer.
- Cons
  - No clean or obvious way to return data to the calling template.
  - Slightly slower than cfinclude.

# Custom Tags

- Pros
  - Real similar to cfmodule (identical actually).
  - Can pass variables in as attributes.
  - Has its own scope, variables are safer.
  - Called in the same fashion as regular HTML tags (and JSP custom tags, etc).
- Cons
  - No clean or obvious way to return data to the calling template.
  - “Return data” construct isn't intuitive.
  - Slower than the other tags we've mentioned.

# cfunction

- Pros
  - Fast.
  - Type-safe arguments and return data.
  - Very clearly shows what arguments get used where.
  - More choices for organizing.
    - Multiple cfunctions in one .cfm file
    - In the same .cfm file as your calling .cfm page
  - CFfunctions are the same as those in a cfcomponent (using them helps prepare you for learning cfcomponents, somewhat).
  - Very similar behavior to functions/methods in other languages (C, C++, Java, etc).

# Cfunction (cont.)

- Cons
  - Requires a 2<sup>nd</sup> step of including the .cfm file first, before you can call the cfunction.
  - If you cinclude the same cfunction twice, CF throws an error.
    - (No “include\_once” construct ala PHP.)
  - Can also leak data.
    - Use the “var” scope!

# What about performance?

- Which construct is faster?
- Technically CFInclude is fastest, but it leaks data, as we saw.
- Custom Tags and CFModule have the same machinery as each other, but are slower than CFInclude.
- CFFunction is a good balance of speed and memory management.
- But the REAL answer is...it probably doesn't matter. (Profile your app first, *then* optimize.)

# Other Thoughts

- If you find you're “shoe-horning” code into a particular construct, take a step back and consider an alternative.
- What's a “best practice” for others may not be best for you (though that is probably a rare exception).
- If you find you're “mixing and matching” lots of these, consider using a framework (e.g. Fusebox, Ben Forta book sample FW).

# Resources

- [cfcustomtags.com](http://cfcustomtags.com)
- [coldfusionjedi.com](http://coldfusionjedi.com) “Custom Tag Tips”
- [coldfusionjedi.com](http://coldfusionjedi.com) “CF Resources”
- [cflib.org](http://cflib.org)
- [fullasagoog.com](http://fullasagoog.com)
- [coldfusionbloggers.org](http://coldfusionbloggers.org)
- BACFUG mailing list
- Ben Forta “WACK” books

# Questions?

Thanks!

I can be reached at...

nolan.erck@gmail.com  
southofshasta.com/blog  
twitter: @southofshasta