

# Another Fun Framework

Posted At : June 1, 2010 9:15 AM | Posted By : Steve

Related Categories: Frameworks, ColdFusion, Neptune

With all of the ColdFusion frameworks that have been released this year, this feels a little like "me to". The fact of the matter, however, is that this framework has been in development for a few years and is being used on a dozens of sites. Moreover, it is different from other ColdFusion frameworks in some pretty significant ways.

The framework doesn't have a name yet. I have been calling it "AFF" ("Another Fun Framework" or "Anti-Frameworks Framework") so far, but it needs a better name pretty soon. I'm open to suggestions, of course.

I wasn't sure how to approach my first blog entry on the framework, so I decided to just answer the questions that I normally ask when I see a framework.

## What is Different about This Framework?

The most obvious difference between AFF and most other ColdFusion frameworks is that it isn't **hub-and-spoke**. That is, you need not have to have all requests routed through one file like "index.cfm". This makes it easy to transition from a "vanilla" ColdFusion application to one using this framework.

The framework also doesn't enforce (or even encourage) a true OO paradigm. It does encourage what **Brian Meloche** referred to as "**OI**" (**Object Inspired**) programming, which follows many of the principles of OO without requiring a deep understanding of OO. Significantly, it assumes the use of queries and structures rather than "objects".

## Why Should I Use This Framework?

The documentation provides some of the goals of the framework, but mostly I would say that you should use it because it does a whole bunch of the tedious work for you.

Although the framework does have an associated **code generator**, the framework itself does most of the work for you by reflection. The result is that CRUD apps have very little code at all and more complicated applications are cleaner because the mundane work doesn't take any visual space in the code. So, the code should more closely match the intent.

Moreover, installing programs onto a site is a snap. Simply copy the folder for a program onto a site and browse to it using any browser. That will install any tables, columns, and required data on to your site. It will also map any included components on the site and set up any needed configurations and admin links as well as run any custom installation code. In short, it will be ready to go.

## How Stable is This Framework?

This is the first Alpha release of the framework, so expect bugs. The framework has been in use for a couple of years and is running on a few dozen sites. Even so, it is still in a pretty constant state of change, so expect a bumpy ride.

I am writing about the framework today more because I gave myself a deadline of June 1 to do so than because of any state of readiness in the framework itself.

Even accounting for time to learn the framework and time to deal with bugs, I expect that most projects will break-even in time using this framework compared to most other options. After that, it should really start paying good dividends. Then again, I'm biased.

## How Can I Get Started?

Although the framework is still in Alpha, it does have a few resources to help with its use.

First, it has quite a lot of [documentation](#). It is far from complete (both in terms of content and in terms of design and functionality), but should still be a helpful resource. I think the "Getting Started" section (patterned after [Dan Wilson's](#) brilliant "So you want to build a ModelGlue:Unity application?" series) is particularly informative.

Second, it has an associated [code generator](#). This can be used to see what code would look like for a provided data structure (good for evaluating the framework) or it can be used to kick-start a project (if you are ready to get going). It has a few example programs included as well, but I wouldn't put much trust in them yet.

I plan to blog about programs using the framework in the coming weeks and months. These will be used as example implementations of the framework, but should also be ready-to-use programs in their own right. I'm hoping to have each one of them be [Jedi-quality programs](#) (a high bar, I know).

If you are interested in trying out the framework, feel free to [download it from the documentation](#) or simply [create a site with the generator](#).