

Hello Swiz...

Posted At : September 18, 2008 1:16 AM | Posted By : Jon Messer

Related Categories: Flex, ColdFusion

Well apparently I'm one of the only people around that really doesn't mind [Cairngorm](#). In fact I'd say I quite like it. It has served me well for the last couple years, and when you add first class IOC like [Prana](#) you have a well documented, flexible, lightweight and battle tested MVC.

Proper architecture avoids the ModelLocator == global variables == bad problem. I tend to only have the highest level layout containers even aware of the ModelLocator, and they bind properties local to the view on to the actual model.

Boilerplate code? Only if you choose to do that, true most of the examples out there do the whole 1:1 event, command, delegate, service thing, but that's a choice left to you as a developer. And Prana has some built in utilities to help mitigate even that boiler plate (no need to write controllers or service locators, delegate factories? no problem...)

My only real gripes are that you can only (easily) have one listener per event. Yes there are ways around this, but they aren't baked into the framework, so you either have to chain commands, have "super" commands that dispatch a series of events, or else you can define multiple controllers and register different commands for the same event but let's not go there...

My other gripe is that in order to have a view notified of the results of an event you have to either piggy back a callback function along with the event or implement an observer on the ModelLocator. Both of which work, but are inelegant.

I evaluated PureMVC a while back but it looked like a more complicated and proscriptive version of Cairngorm. So I wasn't all that motivated to make a change. I also developed a couple apps with ModelGlue:Flex, which I liked, but it seems to have been abandoned, and although it allowed multiple listeners per event, I found it had almost as much boiler plate code and was nearly as verbose as Cairngorm.

All in all, I've been quite happy and productive, so when all the buzz started flying around about [Mate](#) and [Swiz](#) after cf.O I was like yeah OK, IOC? I've had that for quite a while now... MVC? Yeah got that too (it took some work arounds but still...)

So even though I am not unhappy with Cairngorm and I am very happy with Prana, a [couple](#) of [recent](#) posts got me to thinking that I should be giving these new kids a real look. And when the opportunity showed up here at work in the guise of evaluating Flex frameworks to standardize on for some future projects, I had the perfect excuse to give them a look

I haven't gotten to Mate yet, but Swiz... well what can I say?

Dead simple, very flexible, extremely light weight... I like it.

There was a [minor challenge](#) right up front. It didn't work out of the box with the built in Coldfusion remoting end point. But luckily within about 5 minutes of opening up the source code I was able to add the support I needed. Swiz is that straight forward. (I don't know if you've looked at Cairngorm, PureMVC or Prana source trees but they are not what I'd call "simple", they take some grokking before you could really extend or modify them).

So on to the evaluation...

The core principle to Swiz (as I see it) is this... write some cohesive components and classes then wire them together based on annotations in your code. There are also a couple of utility classes to help with dynamic remote objects and a central event dispatcher that works with built in Flex events. But the principle is simple enough and well executed.

What does this look like? Well there is a sparse but easy to follow and complete wiki [here](#). And if you stay tuned here there should be a step by step example of getting up and running with Swiz coming soon...