


Head First Mach-II

Peter J. Farrell
GreatBizTools, LLC

Who is he?

- Co-host of the ColdFusion Weekly
- Lead Developer for Mach-II
- CIO for GreatBizTools, LLC
- Classically trained musician



Good conquers evil
unless
good is stupid

Our Mach-II Goals

I want you to be able to answer by the end

- What is a framework?
- What is type of framework is Mach-II?
- How does Mach-II request lifecycle work?



Things To Remember

- There is no such thing as a stupid question
- Hopefully there are no dumb answers
- You'll hit your head against the wall
- Be willing to take baby steps
 - Iterative process
- Things don't happen over night

OO is hard!



Part One

Jumping into it!



What is a framework?

Programmer speak...

- A core group of files that provide a subset of reusable functionality
- A consistent and logical way of creating and organizing application components
- Tools and/or methodology that allow application components to communicate with each other
- A ***strongly encouraged*** architectural pattern

What is a framework?

English version...

- A framework makes life easier
 - Handles the basic “plumbing” of an application
 - Lets you think less by letting the framework take care of things you do over and over
- A good framework
 - Is easy to learn (relatively)
 - § This does not including learning about OO
 - Helps and doesn't hinder

What is Mach-II?



What is Mach-II?

Part One

- An **event-driven** framework based on the concept of **implicit invocation**
- An **object-oriented** framework and encourages **highly cohesive, loosely coupled components**

What is Mach-II?

Part Two

- Is built on **Model-View-Controller (MVC)**
 - A time tested architectural pattern
- Mach-II is not a full-stack framework
 - Designed to be easily extensible through the plugin and filter architecture

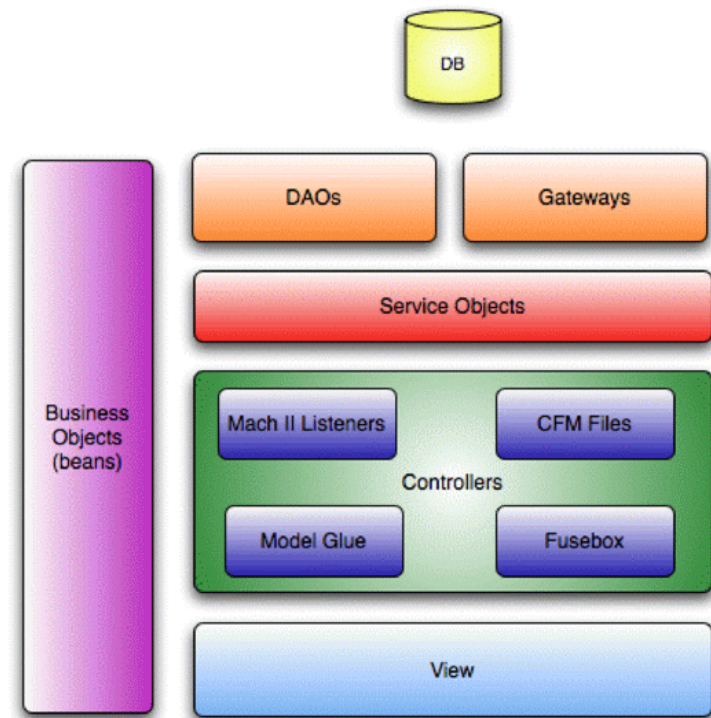
High Cohesion / Loose Coupling?

- **Cohesion:** the degree to which something does one thing well
 - High cohesion = good / Low cohesion = bad
 - I.e. A coffee maker shouldn't make you eggs
- **Coupling:** the degree to which components are dependent upon one another
 - Loose coupling = good / Tight coupling = bad
 - I.e. Your coffee maker needs electricity



What is MVC?

- **Model:** Business objects
 - CFCs
- **View:** User interface
 - CFML/HTML pages
 - AJAX calls
- **Controller:** “traffic cop”
 - index.cfm / framework
 - user-extended CFCs



Part Two

The Wide World of Mach-II



What Happens?

- Mach-II is **event-driven**
 - At the heart of Mach-II is the event object
- The request lifecycle pattern:
 - An event is **announced**
 - “Things happen” (more on this later)
 - Either additional events **announced** or the request ends



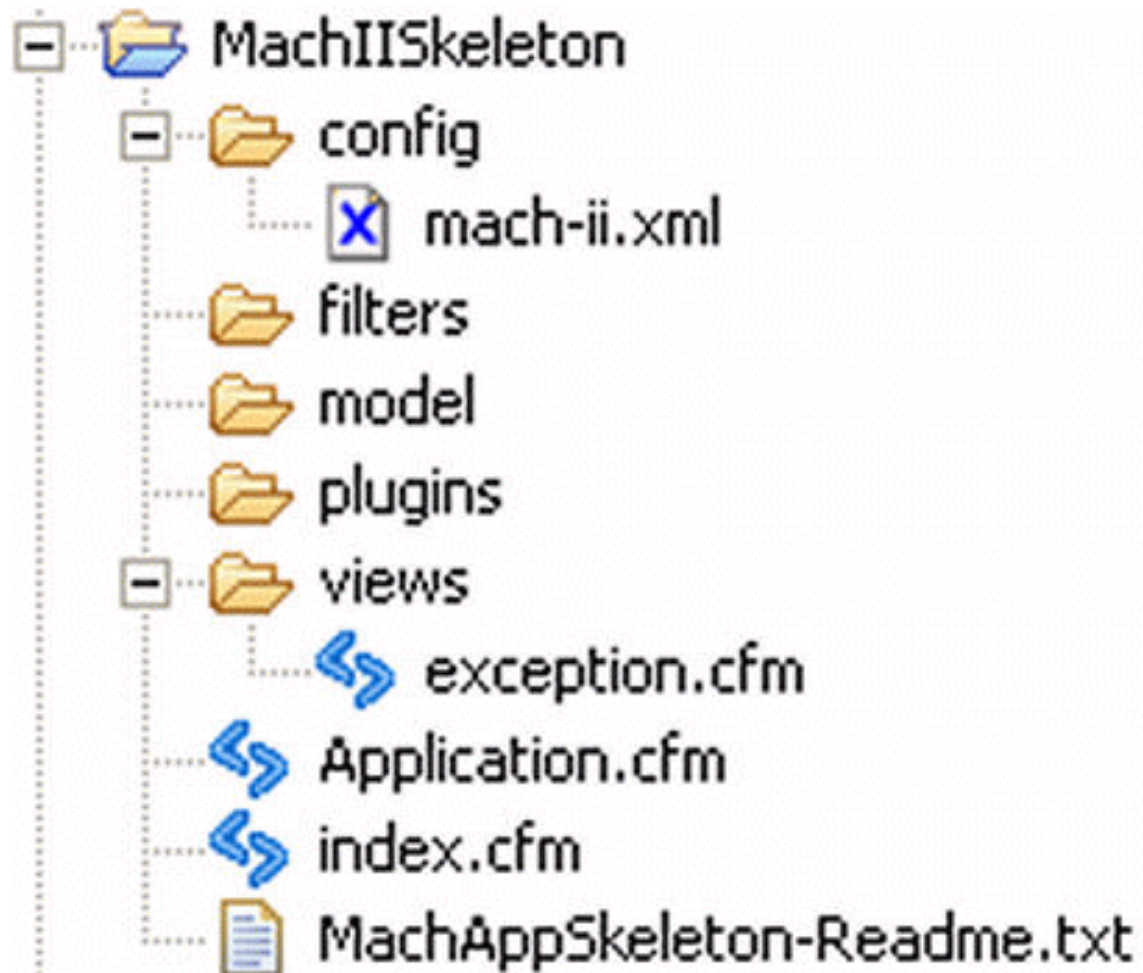
What does the “II” mean?

Mach-II is based on implicit invocation

- Ahah! Mach-*II*...
- Implicit invocation is anti- procedural
- An event announcement **implicitly** causes other “things” to be invoked
- The event does not invoke procedures directly, but announces its’ presence



The Skeleton



A bit about events

- A request is made
`index.cfm?event=buildMonkey`
 - All Mach-II requests are routed through the `index.cfm` (a.k.a front controller pattern)
- The “event” URL parameter corresponds to some code in our Mach-II XML configuration file (more later)

Mach-II Request Lifecycle

- An event is announced
 - In this case, via the URL
`index.cfm?event=processLoginAttempt`
- “Things happen”
 - In this case, the LoginListener is notified of the login attempt and the
`processLoginAttempt()` method is called
- Either another event is announced or the request ends

Part Three

The Request Lifecycle Walk-Through



Our Form Page

- Our form has two fields:
 - Username
 - Password
- Mach-II takes both the Form and URL scopes and injects them in the event object
 - Like a glorified structure/container that holds all form/url variables

Step One

Where it all begins

- The login process starts with a form post to `index.cfm?event=processLoginAttempt`
- The `<event-handler>` command in the Mach-II XML configuration file defines what “things happen”



The `<event-handler>` tag

event-handler tag

event name

access modifier

```
<event-handler event="processLoginAttempt" access="public">  
  <notify listener="LoginListener" method="processLoginAttempt" />  
</event-handler>
```



name of listener

method to call

Step Two

The listener is notified

- Listener “listen” for “notifications” from events
- The framework “notifies” listeners via the `<notify>` when an event involves that listener
- Methods are called within listeners



Login Listener

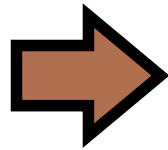
- Yes, it listens for login stuff!
- We'll examine the **processLoginAttempt** event, which is called when will attempt to login a user



Our Story So Far...



Event is announced



Listener is notified

Step Three

- `processLoginAttempt()` takes the event object as its argument
- We get the username and password from the event object
 - `arguments.event.getArg("username")`
 - `arguments.event.getArg("password")`



Validate the Login

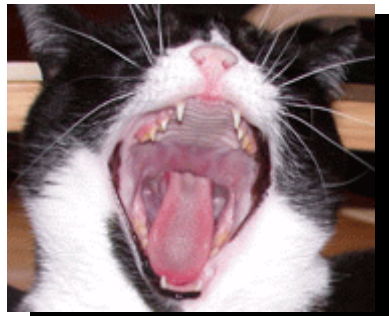
- When have the username and password from the event object, we can validate against any persist source such as a database, LDAP, etc.
- We'll walk through the code in MachBlog...



Step Four

Announce another step

- After the login attempt is processed, the listener announces another event
 - `announceEvent("loginSucceeded")`
 - `announceEvent("loginFailed")`
- These events are **PRIVATE** so they cannot be accessed directly via an URL



Our Pattern Review

- Events are announced
- Data gets injected into the event object
- Listeners are notified
- “Things happen”
- Another event is announced or the request ends

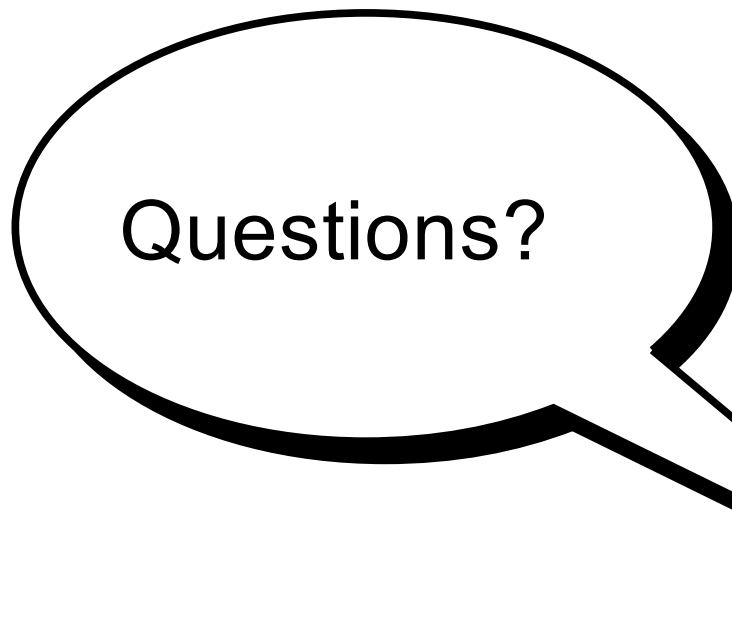
Part Four

Resources and Questions

Resources

- Mach-II Website
 - Including core files, doco, FAQs, samples
 - [Http://www.mach-ii.com](http://www.mach-ii.com)
- Mach-II List at Google Groups
 - <http://groups.google.com/group/mach-ii-for-coldfusion>
- Mach-II.info
 - [Http://www.mach-ii.info](http://www.mach-ii.info)
- MachBlog
 - [Http://www.machblog.org](http://www.machblog.org)

Questions



Contacting Me

- Peter J. Farrell
 - pfarrell@greatbiztools.com
 - [Http://blog.maestropublishing.com](http://blog.maestropublishing.com)

