



# Head First Mach-II



**Matt Woodward & Peter J. Farrell  
cf.Objective() 2006**

# About Us



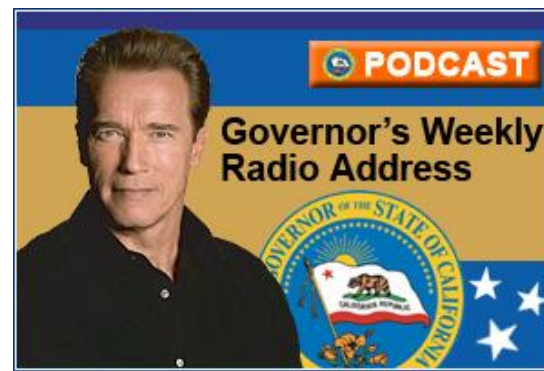
## **Matt Woodward**

- Technical Architect and Lead Developer, Realty InfoLinks in Dallas, TX
- Release Coordinator for Mach-II 1.1.0
- Adobe Community Expert for ColdFusion
- Develop in ColdFusion, Flex, Java, PHP (and even a bit of C#, but don't tell anyone)
- Classically Trained Musician

## **Peter J. Farrell**

- Architect and Developer for Maestro Publishing
- Member of Team Mach-II and Top Contributor
- Classically Trained Musician

# About Our Podcast



- Shameless, shameless, shameless plug
- We're starting the ColdFusion Weekly Podcast next week
- <http://www.coldfusionweekly.com>
- Skype Us! cfweekly
- Call Us! 469-293-3820

# Agenda



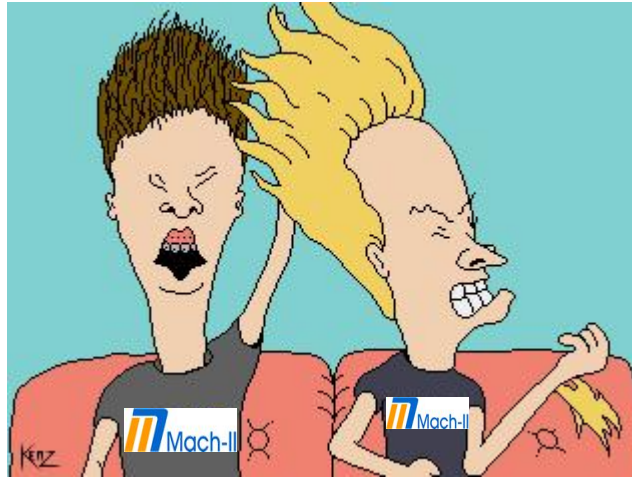
- What's a framework?
- What's Mach-II?
- How does Mach-II work, or, what's Mach-II's "world view"?:
  - Model-View-Controller (MVC): don't worry, it's really quite simple
- "Events, Listeners, and the Framework That Loves Them"
  - A day in the life of the Mach-II request lifecycle
- Looking for a few good developers ...
  - Yes, we will need volunteers from the audience. Seriously.
  - Yes, there will be a ball involved. Seriously.
- Very simple sample application (and I do mean VERY simple)
  - We'll build the login piece, the rest is up to you!

# Goals



- When you leave here you should be able to grab a friend and tell them:
  - What a framework is
  - What Mach-II is
  - What the “II” in Mach-II stands for
  - How Mach-II works
  - How to build a simple Mach-II application (you are allowed to use a writing implement and cocktail napkin)

# Let's Mach(-II)!



- Remember: ***THERE ARE NO DUMB QUESTIONS!!!***

... and hopefully there are no dumb answers either!

# What's a Framework?



- A core group of files that provide a subset of reusable functionality to the language in which the framework is written
  - Can be thought of as an application with an associated API inside which your application runs
- A consistent, logical way of creating and organizing application components
- The tools/methodology that allow the application components to communicate with one another
- A *very strongly encouraged* design pattern (in Mach-II's case, this is MVC)

# What we say to ~~dogs~~

A framework is a group of core files  
you use to build your applications!  
Mach-11 uses CFCs and MVC!

**coders**



# What they hear

blak blak framework blak blak  
Mach-11 blak blak MVC  
blak blak CFC blak blak



# What's a Framework?



- Let's try that again ...
  - Frameworks are here to make our development lives easier by handling the basic “plumbing” of the application
    - Less rethinking the same things over and over, less doing the same things over and over
  - A good framework is (relatively) easy to learn
  - A good framework helps, doesn't hinder

# What's Mach-II?



- Mach-II is an ***event-driven*** framework based on the concept of ***implicit invocation***
- Mach-II is an ***object-oriented*** framework and encourages ***highly cohesive, loosely coupled components***
- Mach-II is built around the ***Model-View-Controller (MVC)*** design pattern

# What's Mach-II?

- o I think I get it!



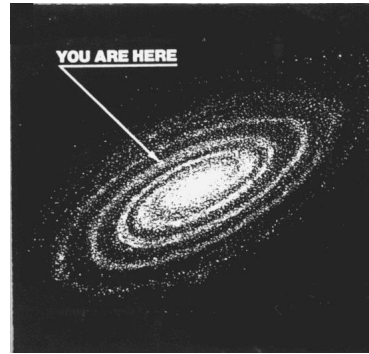
# What's Mach-II?

- Wait a minute ...



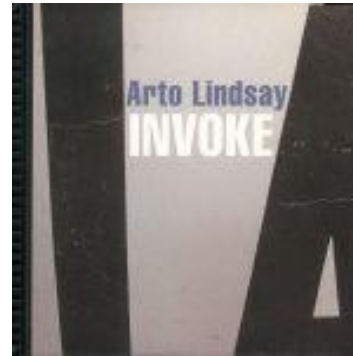
... could you  
repeat that?

# Mach-II's “World View”



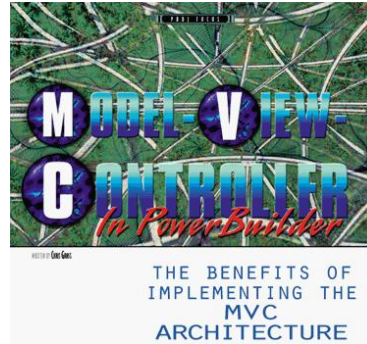
- Mach-II is ***event-driven***
  - The event object is at the heart of Mach-II
  - In a Mach-II request:
    - An event is announced
    - “Stuff happens” (more on this later)
    - Either additional events are announced or the request ends

# Mach-II's “World View”



- Mach-II is based on the concept of ***implicit invocation***
  - Hence the “II”!
  - Implicit invocation is a very anti-procedural or anti-“top-down” approach
  - The event announcement implicitly causes other procedures to be invoked
  - The event does not invoke procedures directly

# Mach-II's “World View”



- Mach-II encourages good application architecture: Model-View-Controller (MVC)
  - **Model:** business objects in the application (CFCs)
  - **View:** pages the user sees and interacts with (CFML pages)
  - **Controller:** “traffic cop” for the application (Mach-II’s index.cfm + framework and user-defined CFCs)

# Mach-II's “World View”



- Mach-II encourages building highly cohesive, loosely coupled components
  - **Cohesion:** the degree to which something does one thing and does it well
    - High cohesion = good; low cohesion = bad
  - **Coupling:** the degree to which components are dependent upon one another
    - Loose coupling = good; tight coupling = bad

# Before We Proceed ...

- Questions?



- Or are we happy Mach-II nerds so far?



# Mach-II Framework Structure

- The Mach-II framework consists of 32 CFCs
  - Developing a successful Mach-II application requires intimate knowledge of each of these CFCs, so let's go through them one by one, line ... by line ... by line ... by line ... by line ...

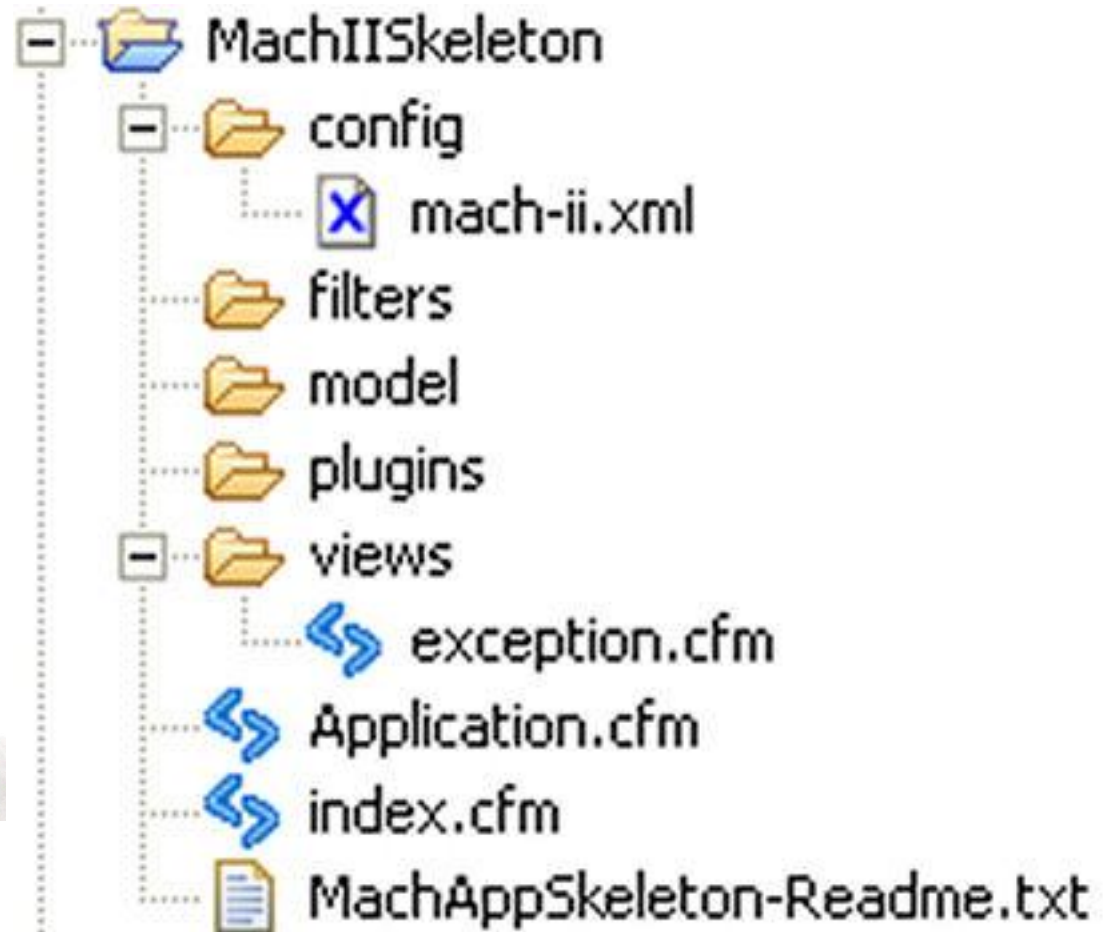


# Mach-II Framework Structure

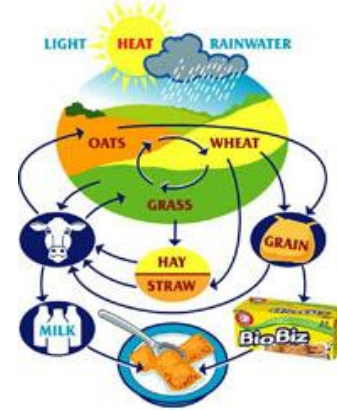


- ***WAIT! DON'T LEAVE!***
- The *GREAT* thing about Mach-II is that you *DON'T* have to worry about the “under the hood” stuff unless you want to!

# Mach-II Application Skeleton

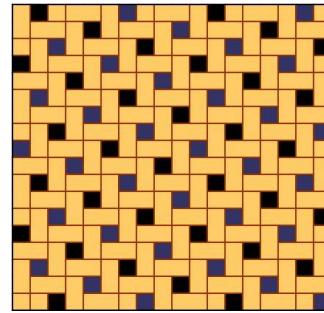


# Mach-II Request Lifecycle



- Typical Mach-II URL:  
index.cfm?event=doSomething
  - All requests in Mach-II are routed through index.cfm
  - The “doSomething” event name corresponds to an event that is defined in Mach-II’s XML configuration file (we’ll see this shortly)


# Remember the Pattern!



## *A sneak preview ...*

1. Event is announced, in this case via the URL  
(`index.cfm?event=processLoginAttempt`)
2. “Stuff Happens,” in this case the LoginListener is notified of the login attempt and the **`processLoginAttempt()`** method is called
3. Either another event is announced, or the request comes to a conclusion

# The Login Form



The screenshot shows a Windows-style dialog box titled "Anmeldeinformationen". It contains a key icon and the text "Geben Sie einen Benutzernamen und ein Kennwort ein, die für dieses System gültig sind." Below this are three input fields: "Benutzername:" (text box), "Kennwort:" (text box), and "Domäne:" (dropdown menu with "URZ\_POOLS" selected). At the bottom are four buttons: "OK", "Abbrechen", "Hilfe", and "Herunterfahren...".

- Where it all begins
- Form fields:
  - username
  - password
- Mach-II grabs both form and URL variables and puts them in the event object

# Step 1: The Event is Announced



- The login process starts with a form post to `index.cfm?event=processLoginAttempt`
- The `<event-handler>` tag in the XML config file defines what happens in this event

# 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

HEY YOU!

## Step 2: Listener is Notified



- Developer-defined objects that “listen” for notifications from events
- The framework “notifies” listeners via the notify command when an event involves them
- Methods (functions) within listeners are then called
  - Typically listener methods will either return something or announce another event

# Sample Listener



Tribune photo by Chuck Baran  
William Holander covers his ears during the performance of the Paul Smoker Trio at the festival, but his wife Dorothy doesn't mind the volume.

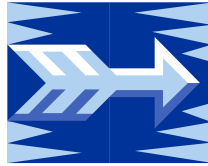
## ○ LoginListener

- Guess what? It listens for “login stuff!”
- We’ll examine the `processLoginAttempt` event, which is called when—you guessed it—a user attempts to log in

# The Story So Far ...



Event is announced



Listener is notified

### 3. Listener Method Called



- The notify command specifies a method to be called, which in this case is `processLoginAttempt`
- The form variables from the login form are passed in the event object itself

# Listener Gets Arguments from Event



- `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")`

# Validating the Login

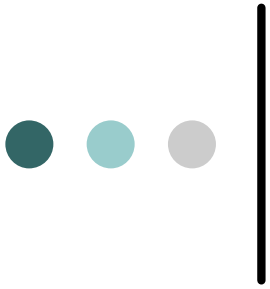


- Once we have the username and password from the event object, we can validate against a database, LDAP, or whatever
  - Already we're seeing tight cohesion/loose coupling come into play

## Step 4: Another Event Announcement

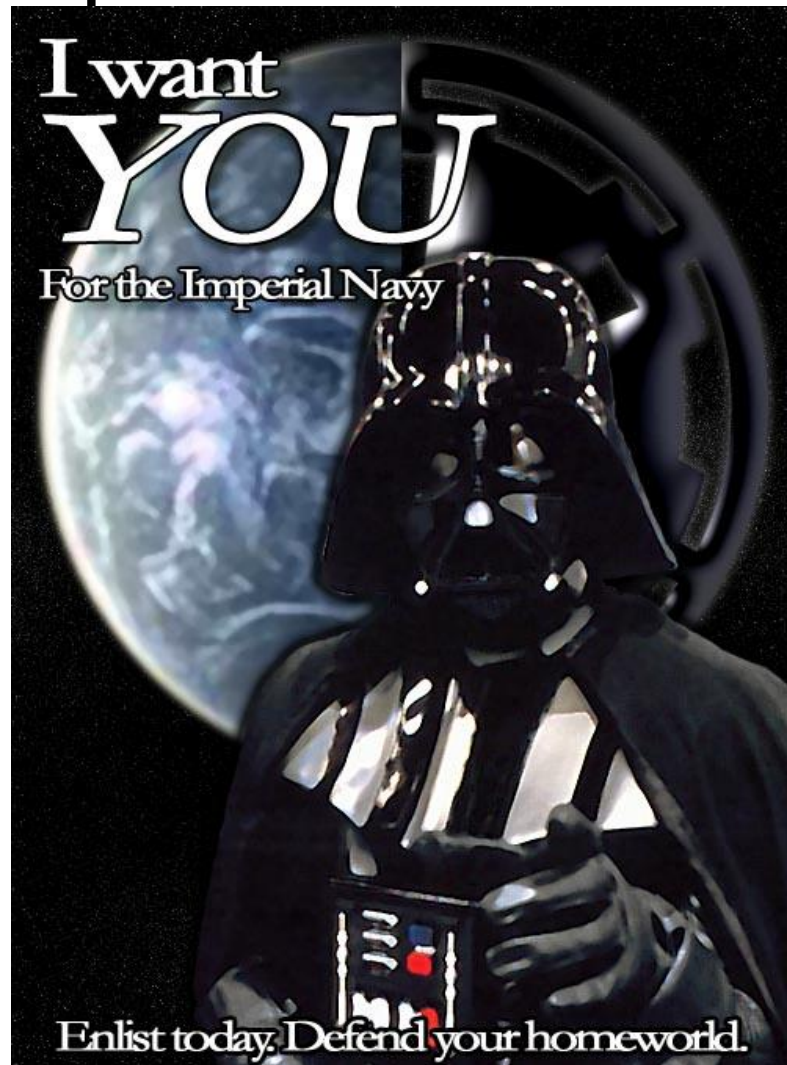


- After the login attempt is processed, the listener announces another event
  - `announceEvent("loginSucceeded");`
  - `announceEvent("loginFailed");`
- These events are PRIVATE so they can't be accessed directly via the URL



**SHUT UP  
AND SHOW  
SOME CODE!!!!**

# Time for YOU to Mach-II



- We need three volunteers!
  - Who's the biggest loudmouth in the room?
  - Who's a good listener?
  - Lastly, any event planners?

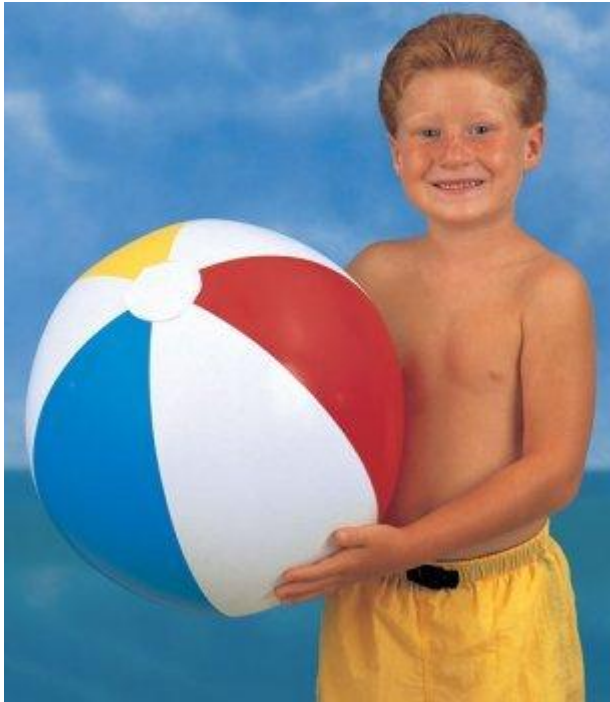
# The Mach-II Players

(think very, very off-Broadway)



- Larry/Lisa Loudmouth, or, the Mach-II Town Crier
  - **Announces** events for the whole framework to hear and **notifies** listeners
- Mr./Ms. Sensitive
  - A great **listener**, and uses any **methods** necessary to get the job done
- The Wedding Planner
  - Keeps track of all the **event** details

# This OO Stuff's a Real Beach

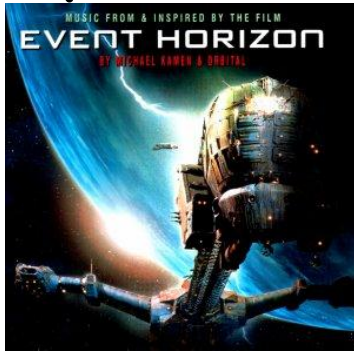


- The event object will be represented by a ball
- As it gets passed from person to person, keep track of what happens!

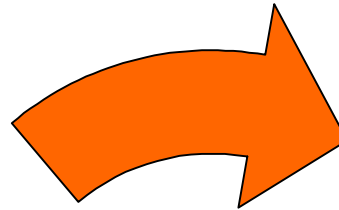
# Event Arguments



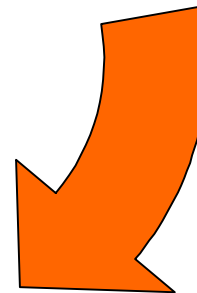
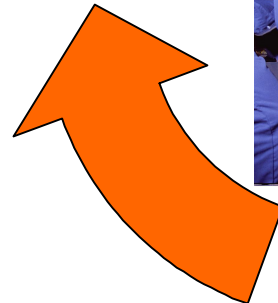
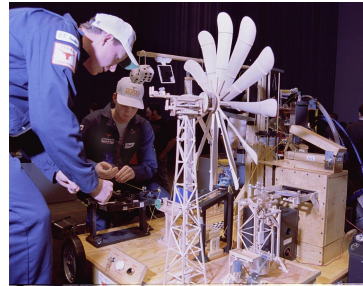
# Once More With Feeling



Event is announced  
(or the request ends)



Listener is notified  
(not mandatory, but typical)



“Stuff Happens”

● ● ● | VOILA!



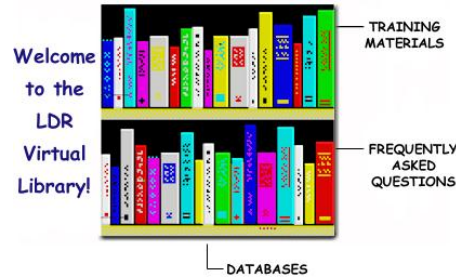
- You just learned how to build a login component in Mach-II!
  - And it was a lot better than finding yourself in a guillotine!

# Review



- Events are announced
- Data gets put in the event object
- Listeners are notified
- “Stuff happens”
- Another event is announced, or the request ends
- ***That’s all there is to it!!!***

# Resources



- Mach-II web site  
<http://www.mach-ii.com>
- Topica List  
<http://lists.topica.com/lists/mach-ii-coldfusion/>
- Mach-II.info  
<http://www.mach-ii.info>
- Sean Corfield's Mach-II Development Guide  
<http://livedocs.macromedia.com/wtg/public/machiidevguide>
- Peter Farrell's FAQs  
<http://blog.maestropublishing.com/>
- My blog  
<http://www.mattwoodward.com/blog>
- Lots of other blogs and resources (Google is your friend!)

# Questions?



- Thanks!
- Matt Woodward  
[mpwoodward@gmail.com](mailto:mpwoodward@gmail.com)
- Peter J. Farrell  
[pjf@maestropublishing.com](mailto:pjf@maestropublishing.com)