MDCFUG
Mike Henke
11/10/2009

Outline: http://bit.ly/2uRcDx
Presentation Slides / Speaker Notes:
http://bit.ly/3zGlrt
Code Examples: http://bit.ly/49tp3t
Recording: add url here

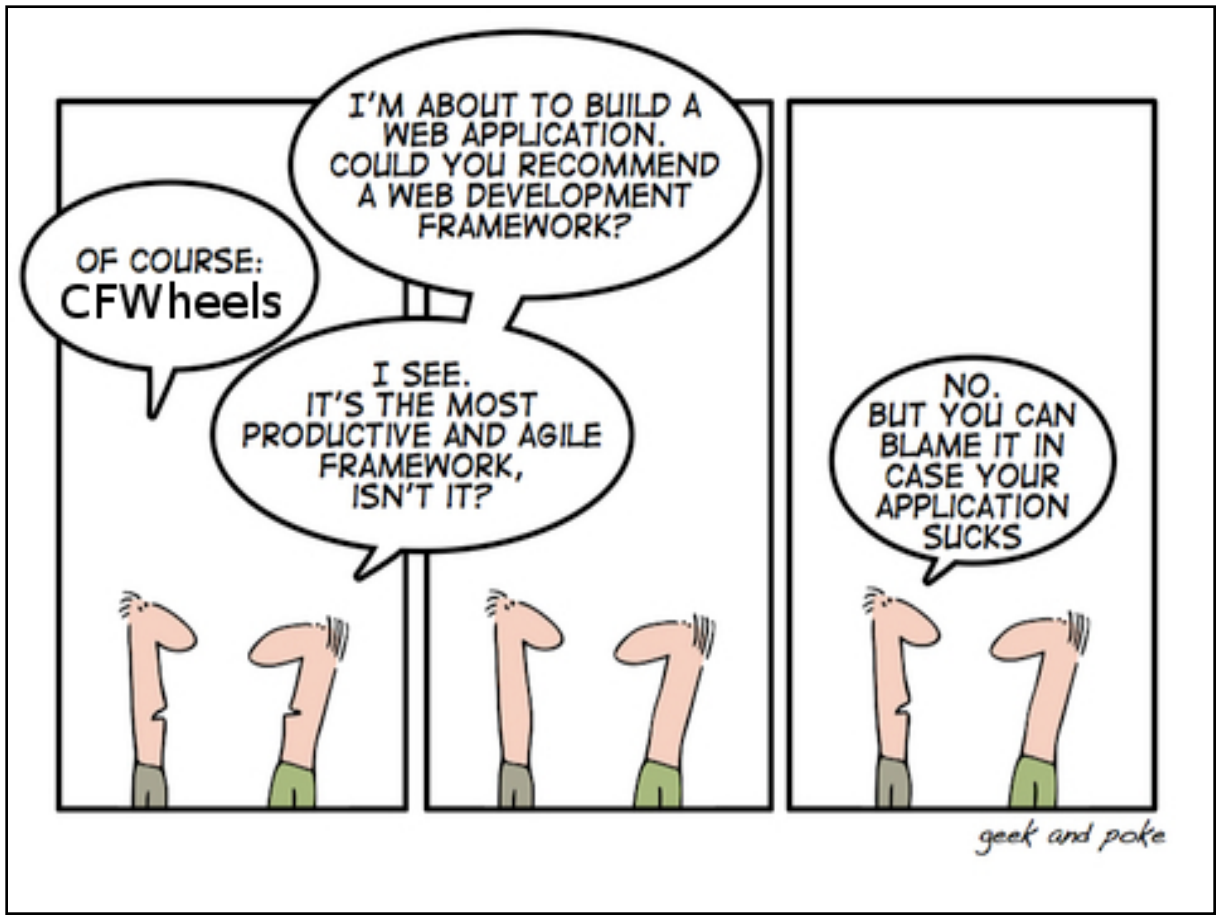
Technical Lead Position
in DC (Metro accessible)
http://bit.ly/Q3M7S

Mike Henke

Work for Acistek as contractor for HHS depart

ColdFusion Developer since 2000

Moved from Omaha to DC area in Jan 09



Do you really need a framework?

- Separate your business logic from your presentation code
- Give you a good code organization structure to follow
- Encourage clean and progrmatic design
- Simplify saving data

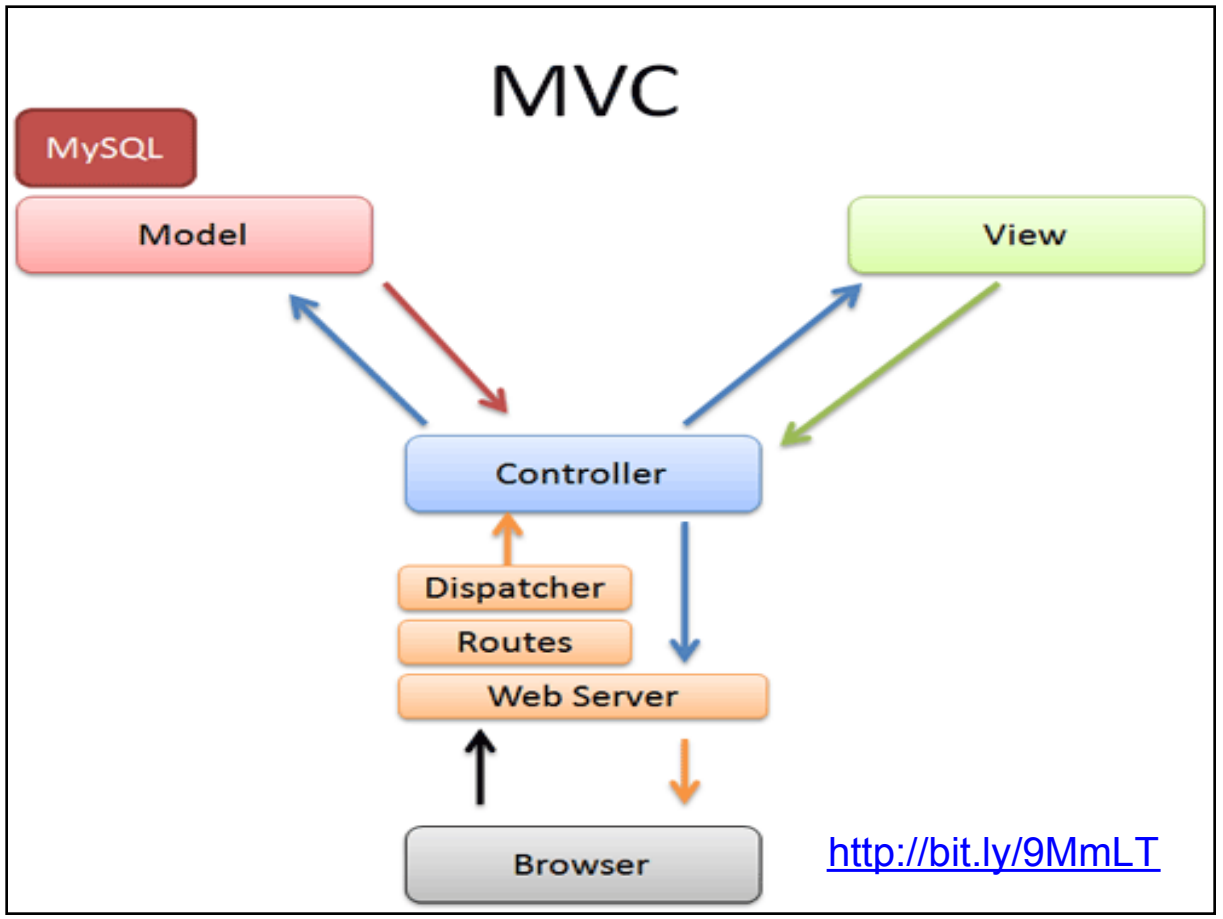
CFWheels is based on RoR but not a direct port so it can better fit CFML.

Target is simple crud websites

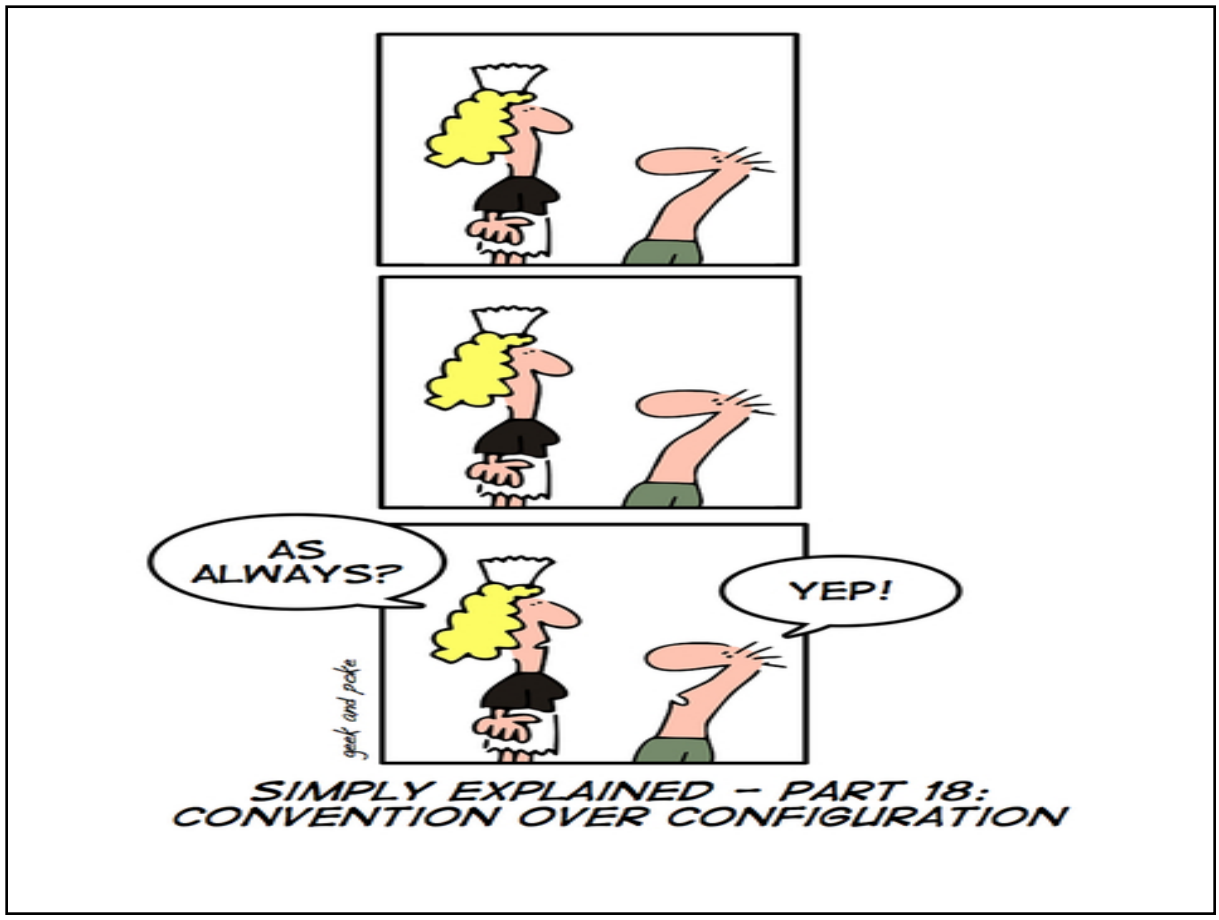
Simpility
- No XML
- No Getters/Setters
- No simple, repetitive SQL

built in ORM - speed up development

Assumes you want to do things a certain way .



- **Controllers The best controller is Dilbert-esque:**It gives orders without knowing (or caring) how it gets done
- **Models** talk to the database, validate data, perform the business logic and otherwise do the heavy lifting.
- **Views** are what the user sees: HTML, CSS, XML, Javascript, JSON. **Views are merely puppets** reading what the controller gives them. They don't know what happens in the back room



Conventions =
Policies =
Default rules

- Naming
- File Locations
  - JS goes here
- Application setting
- Urls
- Calling objects

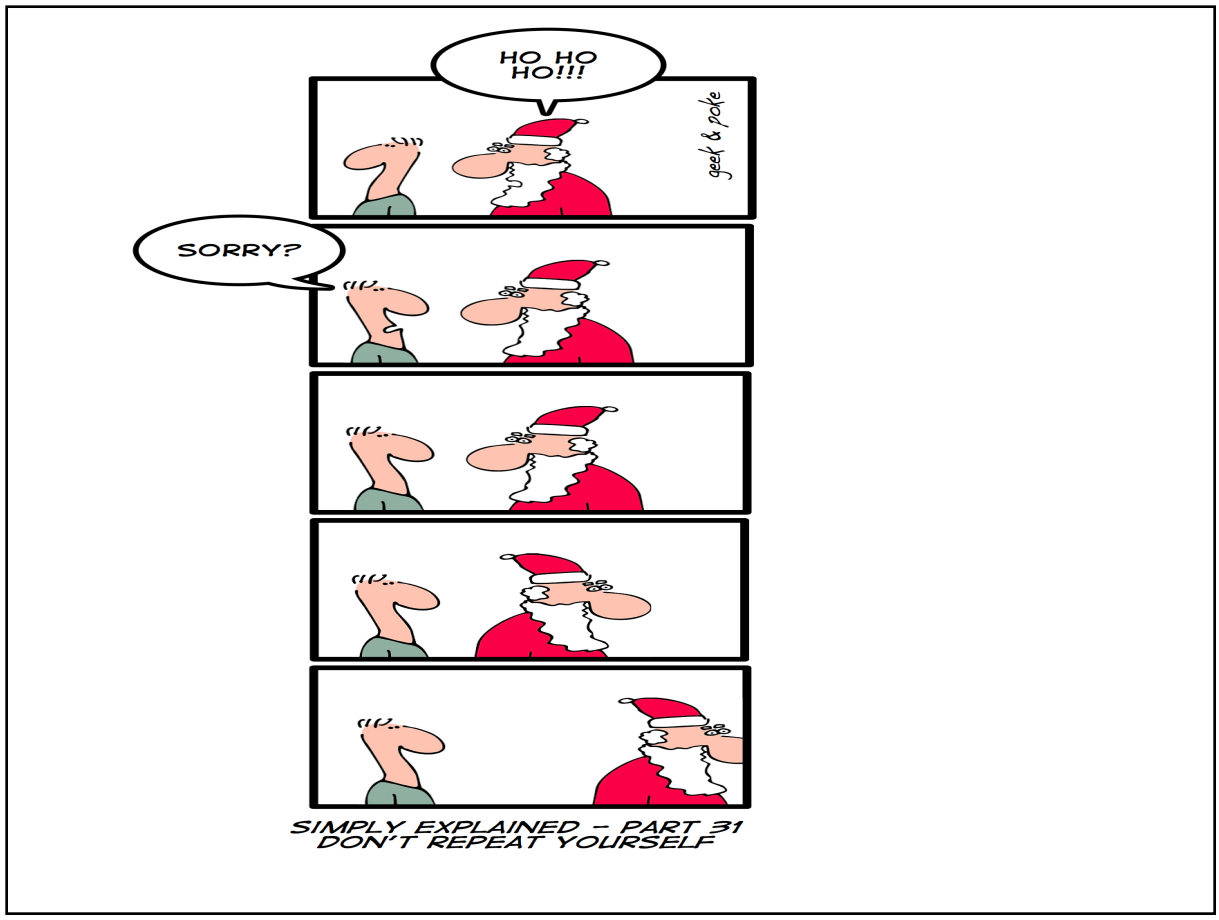
Opininated Software - guides you into their way of doing things

Think of wiki = html to complicated

Only configure extrodinary not ordinary

decrease number of decisions needed, gaining simplicity, but not losing flexibility

sticking to conventions works, keeps things simpler



clear and practical, there is little need to duplicate the information

duplicating, makes changes harder

promotes uniformity, simplicity

Disclaimer:

I, El Henke, take a great career risk, one of the greatest risks taken in Programming Presentations. And that risk is turning over the content portion of this presentation to coding live.

Working With Wheels

Directory structure
- Controllers
- Models
- Views
- Wheels
- Config
- more

Config/Defaults
set() function
- URLRewriting
- Data Source
- Function Settings
- Enviorment
- Roots
- Other Settings

5 environments
different caching and error handling

http://localhost:8309/index.cfm/showing/coc

http://cfwheels.org/plugins

extend Wheels functionality.by using plugins or creating your own.

Scaffolding plugin

Add scaffolding to your Wheels applications. It will enable you to quickly create Controllers, Models and Views.

show working crud

add record, show, update

Handling Requests (controller)

a *controller* takes an incoming request a, decides what (if any) data to get from the model, and decides which view to display to the user.

Request Handing - how incoming requests map to code in your application

http://localhost/index.cfm/shop/products

http://localhost/index.cfm?controller=shop&action=products

shop is the name of the controller to call, and products is the name of the action to call on that controller.

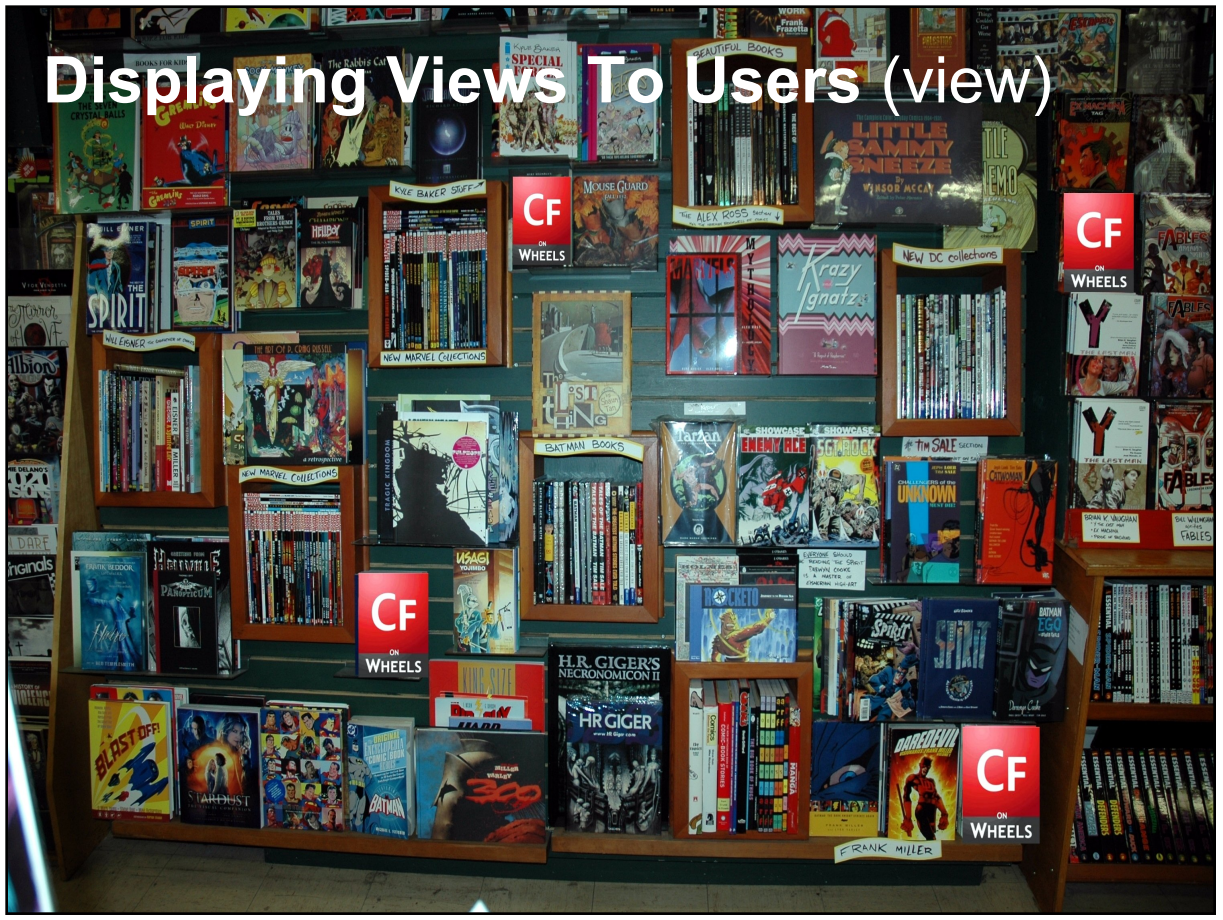
cfdump
params struct combins url and form scope

available to you in the controller and view files but not in the model files

controller/Evals1.cfc
add this to new action

```
<!--- Find the record all --->
	<cfset qualities = model("Quality").findAll()>
	<!--- Create list of level of experience --->
```

Displaying Views To Users (view)



The view renders the contents of a model.

Rendering Content - response to user (controller task)
3 types

- Display Content
  - renderPage() function
- Redirect to another URL
- Send a File

- All views live in views folder
- each controller gets a subfolder named after it in the views folder
- view file to include is a cfml file named after the action

Form Helpers
form with cfwheels conventions

new1.cfm

Partials
#includePartial("xxx")#

- passing in data

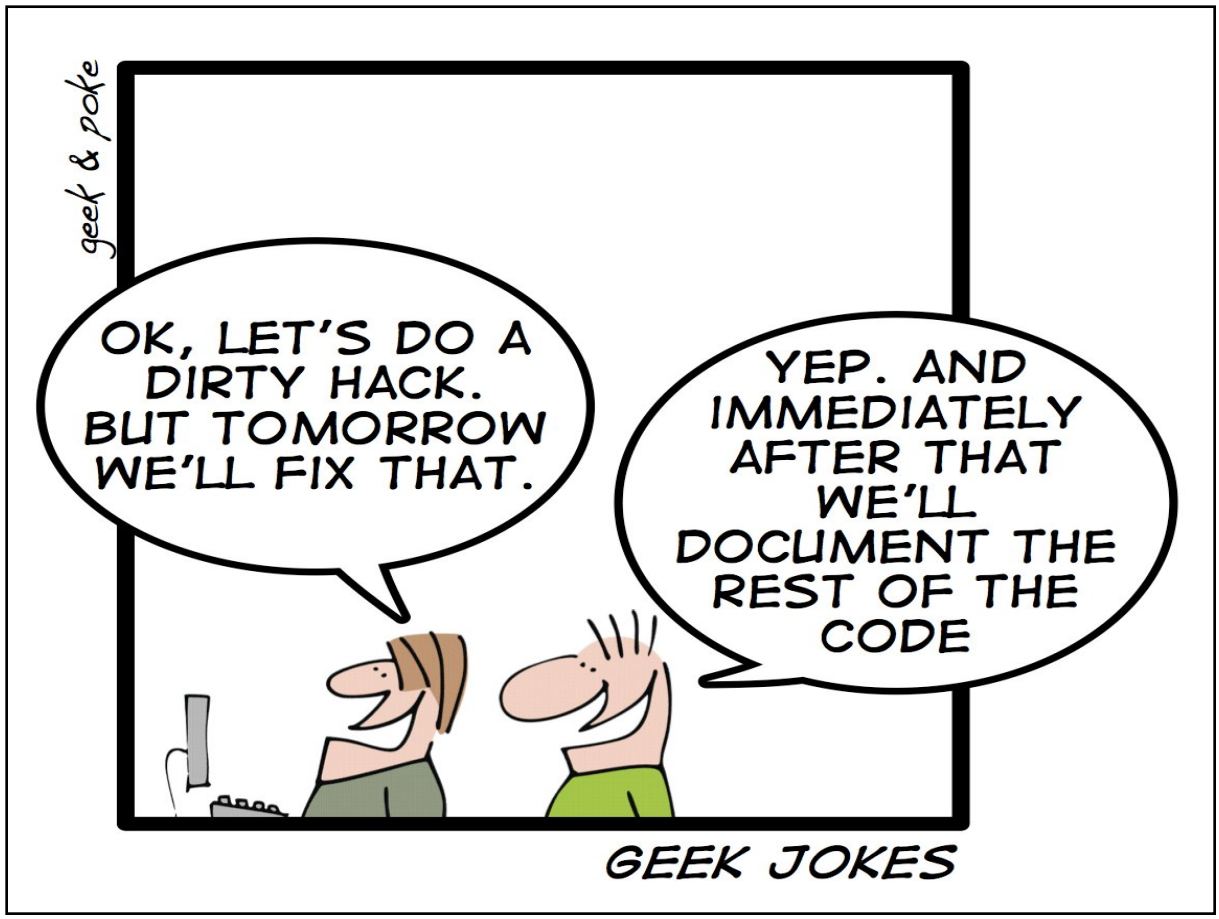
Represents Data

Object Relational Mapping (ORM)
model() function

Naming Conventions:
Tables names plural
Automatic Time Stamps

overriding
<cfset table("tbl_eval") />
column names
No getters/setters properties are available in the this scope.
clean up index1.cfm
model/Eval2.cfc
controller/Eval.cfc
final /controller/Eval.cfc
final /views/index.cfm
associations define the relationships between your database tables

- belongsTo
- hasOne
- hasMany



adding js/css files

adding media helper css, js
layout2.cfm
final index.cfm
flash
final layout.cfm
changing routing


http://cfwheels.org
http://groups.google.com/group/cfwheels
Screencasts by Chris Peters
http://cfwheels.org/screencasts
Head First Rails Applied to CFWheels
http://www.resultantsys.com/index.php/cfwheels
So you want to create a CFWheels application?
http://henke.ws/page.cfm/presentations
DataTables (table plug-in for jQuery) http://www.datatables.net
Cartoons
http://geekandpoke.typepad.com


Now Try Wheels
Or Else
henke.mike@gmail.com
http://www.henke.ws
Technical Lead Position
in DC (Metro accessible)
http://bit.ly/Q3M7S