

FR1D - Process and Tools

# SETTING UP A SOLID LOCAL DEVELOPMENT ENVIRONMENT

Kurt Wiersma



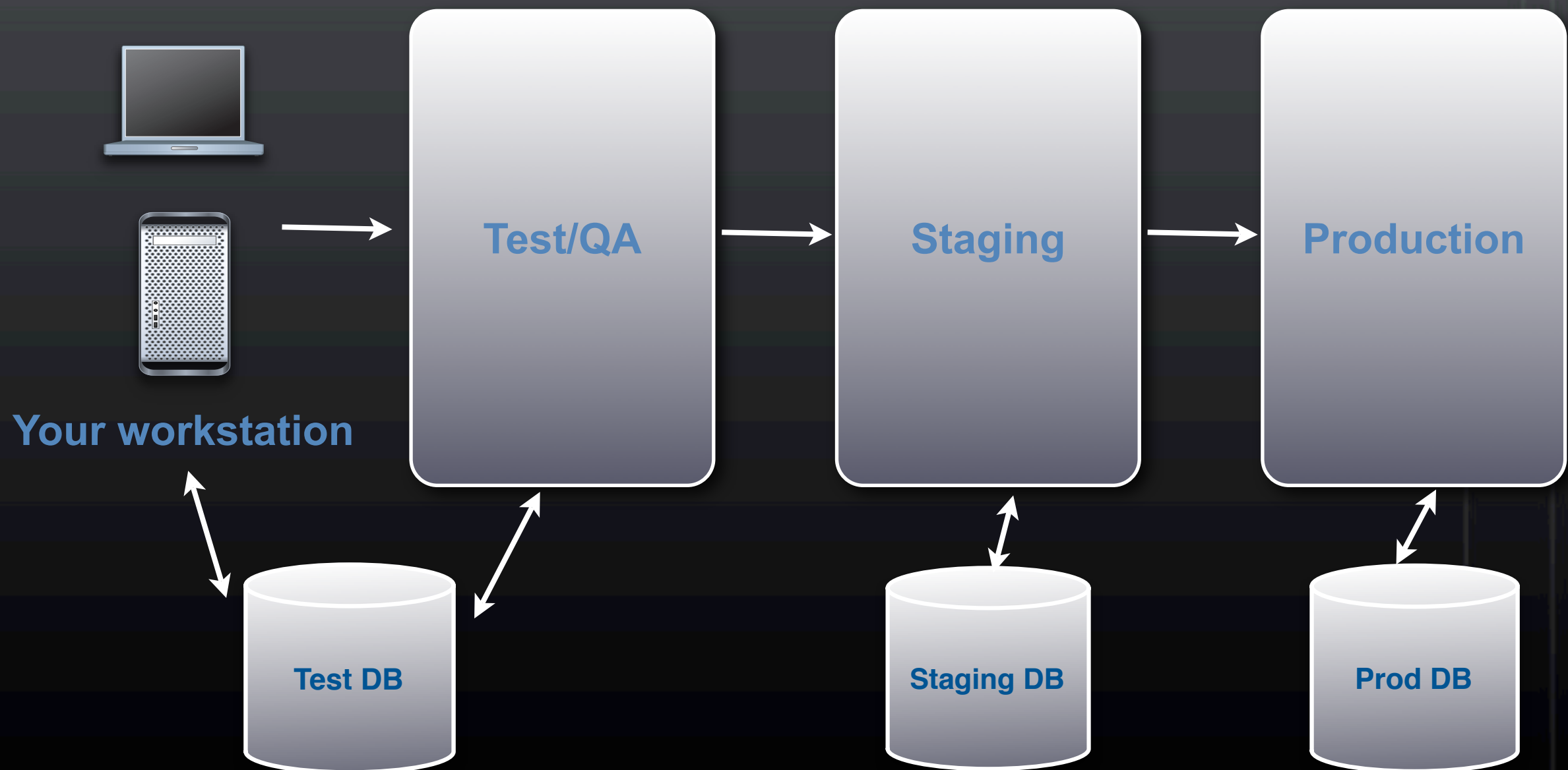
CF.Objective()



# Quick Survey

- Who uses source code control (SVN, CVS, Perforce, ect)?
- Who develops and tests cf code on your workstation?
- Who has CF Enterprise?

# Ideal Environment Setups



# What is a local development environment?

- A “mini” production server on **your** workstation
- A place where you can experiment without worrying about affecting your team members

# Why should we have it?

- Standard development setup across all developers
- Testing locally allows you to experiment
- Overwriting files on a shared server could be disastrous
- The closer your configuration is to production allows you identify changes that will be necessary in production

# Parts of a dev environment

- A web server with virtual hosts
- One or more ColdFusion instances
- Either a local database or a connection to a development database
- A local checkout from source code control
- Development tools to make you productive



# Web Server

- IIS vs Apache
  - Apache used to be easier to manage
  - Choose what you are familiar with
  - Consider using Apache locally even if you deploy to IIS so you can take advantage of virtual hosts
- What are virtual hosts?
- Why not use CF's built in web server?



# Apache With Virtual Hosts

- Configure a new virtual host in Apache
- Update your hosts file
  - Located in `/etc/hosts` on OS X/Linux
  - Located in `/Windows/system32/drivers/etc/hosts` on Windows



# Apache Config

```
5 NameVirtualHost *:80
6 |
7 <VirtualHost *:80>
8     ServerName localhost
9     ErrorLog /var/log/httpd/localhost-error_log
10    CustomLog /var/log/httpd/localhost-access_log common
11 </VirtualHost>
12
13 <VirtualHost *:80>
14     ServerName kurt-cf
15     DocumentRoot "/Users/kurt/Projects/DW Sites/kurtwiersma.com"
16     ErrorLog /var/log/httpd/kurt-cf-error_log
17     CustomLog /var/log/httpd/kurt-cf-access_log common
18
19     Alias /MachII_1_5 "/Users/kurt/Projects/DW Sites/frameworks/machii_1_5"
20 </VirtualHost>
21
22 <VirtualHost *:80>
23     ServerName hematologics-web
24     DocumentRoot "/Users/kurt/Projects/DW Sites/hematologics.com"
25     ErrorLog /var/log/httpd/hematologics-web-error_log
26     CustomLog /var/log/httpd/hematologics-webf-access_log common
27 </VirtualHost>
28
29 <VirtualHost *:80>
30     ServerName local.jidev.com
31     DocumentRoot "/Users/kurt/Projects/DW Sites/www.jidev.com/"
32     ErrorLog /var/log/httpd/local.jidev.com-cf-error_log
33     CustomLog /var/log/httpd/local.jidev.com-access_log common
34 </VirtualHost>
```

# ColdFusion with multiple instances

- Each instance has its own settings, datasources, and mappings
- One instance per client
- One instance per environment

# CF Instance Apache Config

```
135 # JRun Settings
136 Alias /cfide "/Users/kurt/Projects/DW Sites/kurtwiersma.com/CFIDE"
137 Alias /CFIDE "/Users/kurt/Projects/DW Sites/kurtwiersma.com/CFIDE"
138 #LoadModule jrun_module /Applications/JRun5/lib/wsconfig/1/mod_jrun.so
139 # JRun Settings
140 LoadModule jrun_module /Applications/JRun5/lib/wsconfig/1/mod_jrun22.so
141 <IfModule mod_jrun22.c>
142     JRunConfig Verbose false
143     JRunConfig Apialloc false
144     JRunConfig Ignoresuffixmap false
145     JRunConfig Serverstore /Applications/JRun5/lib/wsconfig/1/jrunserver.store
146     JRunConfig Bootstrap 127.0.0.1:51020
147     #JRunConfig Errorurl url <optionally redirect to this URL on errors>
148     #JRunConfig ProxyRetryInterval 600 <number of seconds to wait before trying to reconnect>
149     #JRunConfig ConnectTimeout 15 <number of seconds to wait on a socket connect to a jrun se>
150     #JRunConfig RecvTimeout 300 <number of seconds to wait on a socket receive to a jrun serv>
151     #JRunConfig SendTimeout 15 <number of seconds to wait on a socket send to a jrun server>
152     AddHandler jrun-handler .jsp .jws .cfm .cfml .cfc .cfr .cfswf
153 </IfModule>
```

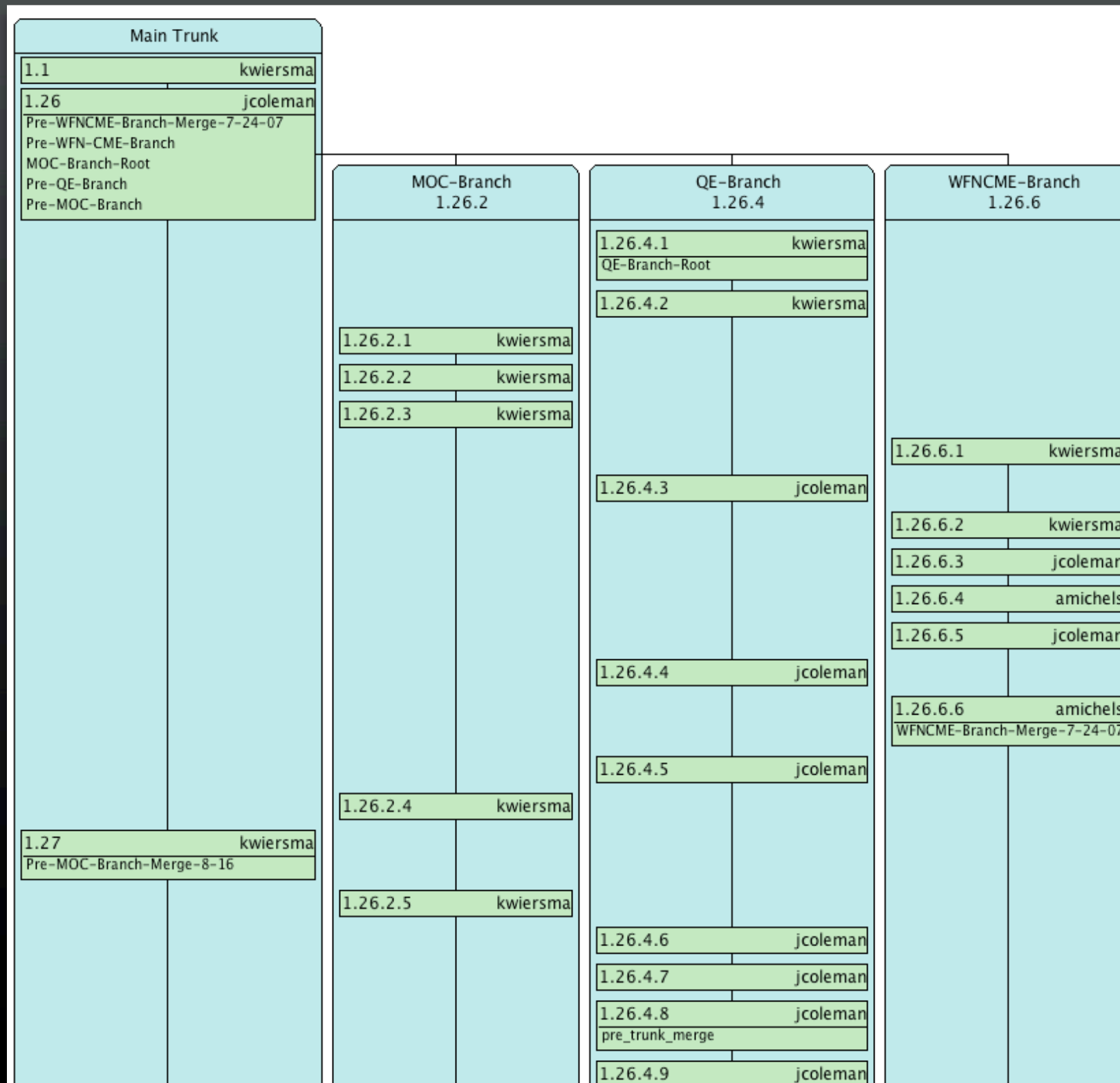
# Development Databases

- Use either a local development database or share one
- MySQL vs. MS SQL
  - You can run SQL Server using VMWare or Parrells
- Pros and cons of a shared development database
- Keeping database in sync across developers
  - DDLUtils

# Source Code Control

- Source code control directory setup
  - Root of your project as the webroot
- Use branches to split out your development projects from the trunk which should match production
- Consider installing Trac, TracFu, or ViewCVS

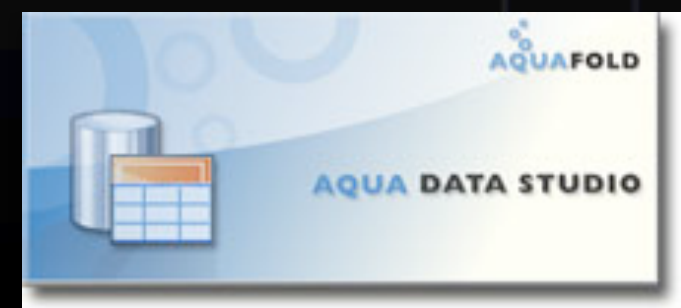
# Branching Example





# Development Tools

- CFEclipse/Eclipse (of course!)
  - Mylyn
- A good source code control client
  - SmartSVN/CVS, TortoiseSVN/CVS
- Apache Ant
- A good database tool
  - Aqua Data Studio, SQL Explorer





# Development Tools

- Firefox Plugins
  - Firebug, Web developer toolbar
- Selenium - functional testing
- CFCUnit

# Selling to your co-workers

- Easy collaboration amongst team members
- No more lost code due to someone uploading files over your changes
- Fewer errors when client are testing
- Portable environment means you potentially can work from anywhere

# Selling to your boss

- Clients will be happier because there will be fewer errors when testing
- Some initial setup time but then developers will be more productive
- Easier to track down issues right after deployment
- Fewer bugs related to bad paths



# Taking it to the next level

- Leverage Ant (Attend Peter Farrell's session)
  - Deployment, unit testing, functional testing, loading your test database
- Selenium for functional testing
  - Record your tests in Firefox and then automate running them with Ant
- CFCUnit (Attend John Mason's session)
  - Run your unit tests before committing your code

# Key Points

- Mirror your production environment as much as possible
- Develop locally use a shared dev server only for integration and customer testing
- Use source code control and branches to assist your development

# Key Points

- In many cases you can use a shared DB server for testing
- Leverage the multiple instance features of the CF dev version
- Use tools to make automate repetitive tasks and to make yourself more productive

# Key Points

- Use an issue tracker to track bugs, feature requests, and perhaps even requirements
- Keep a standard, clean directory structure when ever possible.
- Don't relay on file paths, leverage relative paths or `expandPath`. Use mappings if you have to.



# Resources

- Video - [Installing Apache, SVN, and Trac on Windows](#)
- [Aaron West - Apache, SVN, and Subclipse](#)
- [Apache Downloads with SSL](#)
- [ACME Guide \(Apache, CF, MySQL, and Eclipse\)](#)

# Advanced Resources

- [Ant with CF resources from Jim Priest](#)
- [Selenium with CF from Jim Priest](#)
- [John Mason on CFEclipse, CFCUnit and Selenium](#)

# Questions? Fire Away...

