

ANECDOTAL DEVELOPMENT & DEPLOYMENT

A Case-Study in PHP Project Workflow

Lone Star PHP - June 2011

Brian Blood



PHP WEB DEVELOPMENT

- Specify
- Plan
- Write
- Test
- Approve
- Log
- Repeat

PHP WEB DEPLOYMENT

- Stage
- Deploy
- Bill

SCENARIO

- Object13 Development - www.object13.com
- 3 developers, 1 project manager
- All work done at company office location
- Dev, staging, SCM, logging systems at office
- Production systems at colocation facility
- Mix of Mac/PC developer systems

TECHNOLOGIES

- Mac OS X Server 10.5
- Built-in Apache
- Third Party-built PHP 5.2 (Entropy.ch)
- MySQL.com pre-built MySQL server
(/usr/local/mysql)
- Subversion

DEVELOPMENT WORKGROUP RESOURCE REQUIREMENTS

- Apache / PHP / MySQL environment
- Source Code Management
- Flexible layout for configuration
- Predictable layout
- Use Apache config to solve development vs deployment differences
 - SetEnv SystemType "dev" - in global Apache config on dev server; code can make choices if not running in production
 - Project specific Apache included config files; universal and local versions

OFFICE LAN

- Developers connect to central development server (Dev1) via OS native file-sharing protocol via GigE switched network.
 - AFP Share called Projects mounted on Mac desktops
afp://dev1.object13.com/Projects
 - PCs attach network SMB share as P:
\\dev1.object13.com\Projects
- SVN Server on Dev1
access via: svn://svn.object13.com/
- MySQL server on Dev1, accessed via localhost/socket
- VPN tunnel to colo for secure access to backside of production servers.

WORKING COPY WORKINGS

- All working copies live and are executed by the AMP stack on the central dev server
- Provides a stable, known, common development platform for all developers
- Unique URL for each developer / working copy / project
 - register new domain: object13.com -> object13dev.com
 - `http://dev1.widgetsrus.acme.object13dev.com`
 - easy DNS setup: *.object13dev.com

PROJECTS LAYOUT

- Projects
 - Acme - widget producing client
 - General Documentation
 - WidgetsRUs website
 - WidgetPros website
 - Client B...

PROJECT LAYOUT

- /Projects/ Acme/ WidgetsRUs
 - /docs
 - /logs - Apache access/error logs, PHP error logs
 - /cache
 - common storage location used by all working copies (large files)
 - /dev1
 - working copy for developer Bob
 - /dev2
 - working copy for developer Alice
 - /stage
 - working copy for client testing/ approval

BOB'S WORKING COPY LAYOUT

- Disk: /Projects/Acme/WidgetsRUs/dev1
SVN: /clients/acme/widgetsrus/trunk
 - /admin - setup/maintenance scripts
 - /cache - storage specific to each working copy
 - /config - apache tweaks, default & local config vars, db connection setups
 - /docs
 - /docs/schema - Customers.sql, Widgets.sql
MySQL table Create statement and Alter statement history/comments
 - /inc - source code files
 - /inc/sitecommon.php
 - /inc/startup.php - bootstrap file
 - /inc/classes/ - autoloader starts from this directory
 - /lib - non-source files - fonts/templates
 - /website - Apache DOCUMENT_ROOT points here
 - /tests - PHP test files.

APACHE VHOST CONFIG - STAGE

```
<VirtualHost *:80>
  ServerName widgetsrus.acme.object13dev.com
  DocumentRoot "/Object13/Projects/Acme/WidgetsRUs/stage/website"
  SetEnv SiteRoot "/Object13/Projects/Acme/WidgetsRUs/stage"
  CustomLog "/Object13/Projects/Acme/WidgetsRUs/logs/access.log" combinedhost
  ErrorLog "/Object13/Projects/Acme/WidgetsRUs/logs/error.log"
  <Directory "/Object13/Projects/Acme/WidgetsRUs/stage/website">
    AllowOverride FileInfo
  </Directory>
  php_admin_value error_log "/Object13/Projects/Acme/WidgetsRUs/logs/php-error.log"
  Include "/Object13/Projects/Acme/WidgetsRUs/stage/config/httpd.local.conf"
  Include "/Object13/Projects/Acme/WidgetsRUs/stage/config/httpd.conf"
</VirtualHost>
```

- **SiteRoot** environment variable. Use in bootstrap line at top of PHP files providing location independence:
require_once(\$_SERVER['SiteRoot'].'/inc/startup.php');
- Access, Error and PHP error logs are output into project specific **/logs** directory
- Include of project specific Apache config directives. Rewrite rules, ForceType and Environment variables.

```
httpd.local.conf:
SetEnv AcmeTmp "/Object13/Projects/Acme/WidgetsRUs/stage/cache"
Rewritemap ipblockmap txt:/Object13/Projects/Acme/WidgetsRUs/cache/ipblacklist.txt

httpd.conf:
<Location ~ "^/(robots.txt|sitemap.xml)$">
  ForceType application/x-httpd-php
</Location>
```

APACHE VHOST CONFIG

BOB'S WORKING COPY

```
<VirtualHost *:80>
  ServerName dev1.widgetsrus.acme.object13dev.com
  DocumentRoot "/Object13/Projects/Acme/WidgetsRUs/dev1/website"
  SetEnv SiteRoot "/Object13/Projects/Acme/WidgetsRUs/dev1"
  CustomLog "/Object13/Projects/Acme/WidgetsRUs/logs/access.log" combinedhost
  ErrorLog "/Object13/Projects/Acme/WidgetsRUs/logs/error.log"
  <Directory "/Object13/Projects/Acme/WidgetsRUs/dev1/website">
    AllowOverride FileInfo
  </Directory>
  php_admin_value error_log "/Object13/Projects/Acme/WidgetsRUs/logs/php-error.log"
  Include "/Object13/Projects/Acme/WidgetsRUs/dev1/config/httpd.local.conf"
  Include "/Object13/Projects/Acme/WidgetsRUs/dev1/config/httpd.conf"
</VirtualHost>
```

httpd.local.conf:

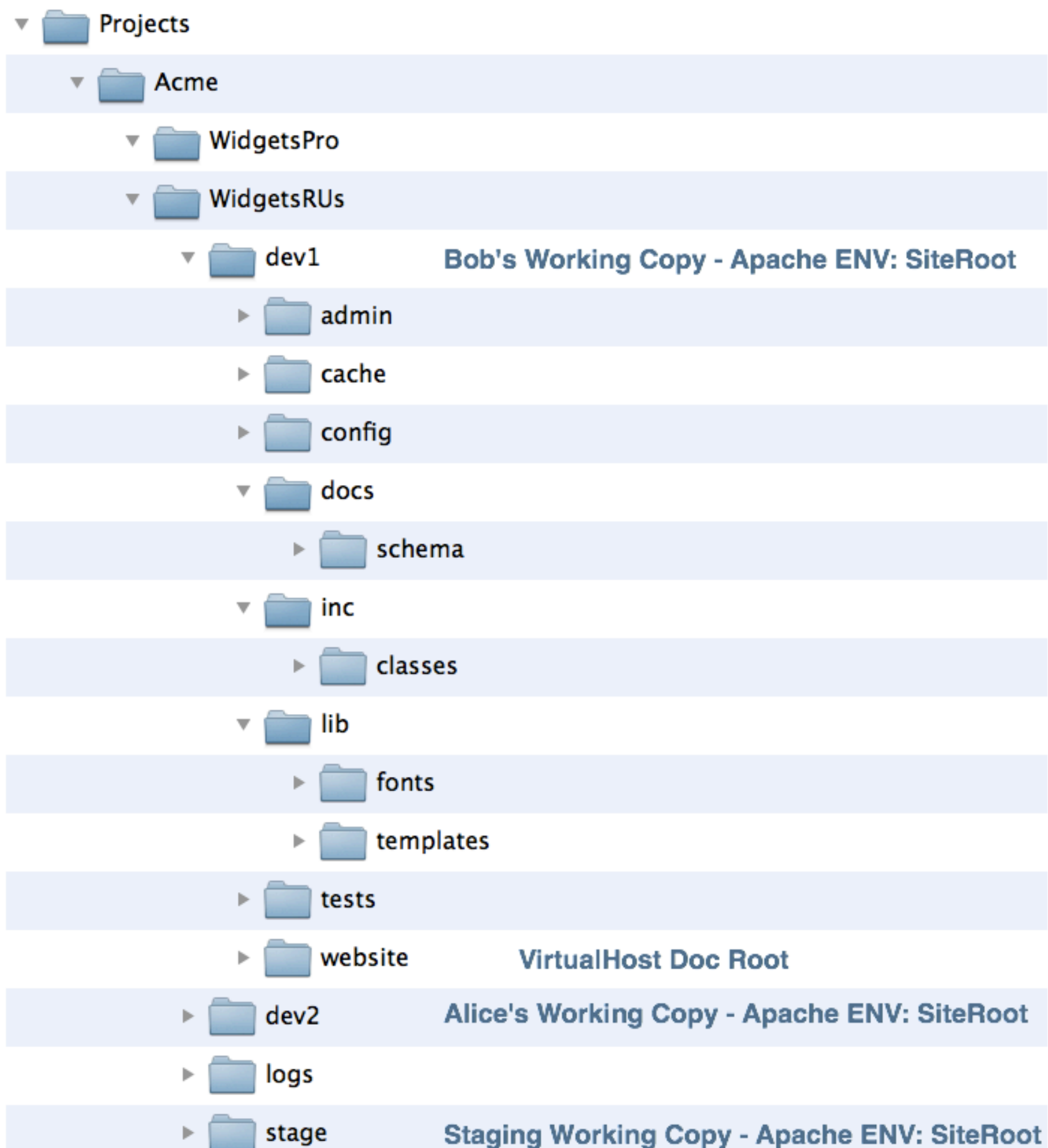
```
SetEnv AcmeTmp "/Object13/Projects/Acme/WidgetsRUs/dev1/cache"
Rewritemap ipblockmap txt:/Object13/Projects/Acme/WidgetsRUs/cache/ipblacklist.txt
```


APACHE VHOST CONFIG PRODUCTION

```
<VirtualHost *:80>
  ServerName www.widgetsrus.acme.com
  DocumentRoot "/Sites/Acme/WidgetsRUs/website"
  SetEnv SiteRoot "/Sites/Acme/WidgetsRUs"
  CustomLog "/Sites/Acme/WidgetsRUs/logs/access.log" combinedhost
  ErrorLog "/Sites/Acme/WidgetsRUs/logs/error.log"
  <Directory "/Sites/Acme/WidgetsRUs/website">
    AllowOverride FileInfo
  </Directory>
  php_admin_value error_log "/Sites/Acme/WidgetsRUs/logs/php-error.log"
  Include "/Sites/Acme/WidgetsRUs/dev1/config/httpd.local.conf"
  Include "/Sites/Acme/WidgetsRUs/dev1/config/httpd.conf"
</VirtualHost>
```

httpd.local.conf:

```
SetEnv AcmeTmp "/Sites/Acme/WidgetsRUs/cache"
RewriteMap ipblockmap txt:/Sites/Acme/WidgetsRUs/cache/ipblacklist.txt
```

APACHE CONFIG BITS

- Added `ht.access` to the `AccessFileName` directive
Files beginning with *period* tend to become invisible in modern file systems. Also add directive to deny serving that file.
`AccessFileName .htaccess ht.access`
- Deny access through web server to `.svn` directories on dev and production.

```
<LocationMatch ".*\.svn.*">  
    Order allow,deny  
    Deny from all  
</LocationMatch>
```


APACHE CONFIG BITS

- Installed Standalone version of the W3C Validator on server for testing output HTML/CSS.
<http://validator.w3.org/>
<http://habilis.net/validator-sac/> - Mac specific installer
- Install self-signed SSL certificate: *.object13dev.com that can be used to configure a SSL version of any site/virtual host being developed. Allows testing switching back and forth between non-SSL and SSL contexts on an eCommerce site.

SITE LAYOUT/APACHE CONFIG SUMMARY

- Use Apache config/environment variables to abstract out differences between each development site and also production site.
- Use Apache config to support a flexible, predictable and organized layout for site. Only expose those items to the web document root which need to be served through the web document root.
- Provide logical, organized directories. A site is **not** just "htdocs".

PROJECTS EXTRAS

- Provide each developer with Sandbox for them to test out new ideas. Use same config / site layout. Predictable!

<http://dev1.sandbox.object13dev.com>

- Common Source code Library. Avoids making copy of ADODB into every project. Version lock-in is a danger.
- Create / Maintain template of Site Layout

PHP INFRASTRUCTURE

- Bootstrap code: (location independent)

```
require_once($_SERVER['SiteRoot'] . '/inc/startup.php');
```

- startup.php:

```
require_once($_SERVER['SiteRoot'] . '/inc/classes/App.inc.php');  
App::Startup();
```

- bottom of App.inc.php

```
function __autoload($class_name)  
{  
    ... autoload logic...  
    class files stored in dirname(__FILE__) or lower directories  
}
```


PHP INFRASTRUCTURE

- App.inc.php methods

- Startup()

load in default & local config vars, load constant definitions, load Session, wakeup stored objects, setup db connections, register_shutdown_function to App::Shutdown, etc.

- Shutdown()

force close db connections, close Session, write log / debug data, etc.

- Config(), SetConfig()

retrieve config var, set config var

PHP INFRASTRUCTURE

- App.inc.php methods, cont'd

- Dir()

- returns full path on disk to a known directory; ex: Template files to load file to process or Fonts for PDF library, Temp files, File Uploads,

- URL()

- returns relative or absolute URL to resource; ex: CSS file to set to load on current page:

- ```
<link type="stylesheet" href="<?= App::URL('main.css', 'stylesheet'); ?>">
```

- Abstraction provides flexibility for where that stylesheet file will actually live.

- In development, the file is served from local working copy:

- ```
<link type="stylesheet" href="/assets/css/main.css">
```

- In production, the file is served from CDN:

- ```
<link type="stylesheet" href="http://acmewidgets.amazons3.com/main.css">
```

# PHP BITS

- /config
  - constants.inc.php - Should contain nothing other than define() statements.
  - crontab.txt - list of cron jobs necessary for site
  - defaults.php - default settings for anything
    - No Magic Numbers!
    - Accessed through App::Config()
  - hostconfig.php
    - overrides settings defined in defaults.php
    - create devsample and prodsample versions to document changes



# SOURCE CODE CONTROL

- Subversion - single repository for all clients / projects. Six of one.....
- Atomic changes. Rarely use tags; 99% goes to trunk
- Hooks
  - Pre-commit - runs PHP syntax check on PHP files
  - Post-commit
    - Send email with files changed / comment to dev team: scmnotify@
    - Auto-Update Staging site with SVN update  
Uses manually maintained lookup table text file of SVN tree to local disk location of where to run **svn up** against.  
/clients/acme/widgetspro --> /Object13/Projects/WidgetsPro/stage

# MySQL

- Schema changes were rare enough to allow as a default policy to connect all working copies to same MySQL database and access same set of tables.
- All queries go through database abstraction layer:
  - DBSelectCell, DBSelectOne, DBSelectList
  - DBInsert, DBReplace, DBUpdate, DBDelete
  - DBQuery, DBFetchRow, DBQuote, DBDisposeQuery

# MYSQL TESTING

- Production MySQL servers replicated back to system at office. Once a day by automated script, a snapshot of the production database is made to create an environment for us to test code against a real-world set of data.
- MySQL development server is always configured with low threshold for slow queries (1 sec) and to log queries not using indexes to help determine potential bottlenecks in table setup.



# DEPLOYMENT SETUP

- Keep Development and Deployment environments as close as possible. Same version of PHP. Same version of MySQL.
- Deployment should be as easy as creating another working copy out of the source code management system. Apache config abstractions should take care of the differences such as location in file system.

# DEPLOYMENT OF SOURCE

- Updates out of SCM should be performed as the same user to avoid ownership and permissions problems. (codeadmin)
- Multi-server updates - "bastion host" that can password-less ssh/svn up commands against all members of load balanced group.
- Create script for post checkout configurations on new systems. mkdir / chmod / chown  
chgrp www /cache; chmod g+wX /cache
- Create script to test deployment setup

# IS IT SAFE?

- Nightly **svnadmin dump** against repository. Output is compressed and uploaded to server at datacenter.
- Local MySQL development server is setup as replication Master. Replication Slave at datacenter keeps copy of development databases offsite.
- Twice Daily **mysqldump**. Tables have been dropped by mistake more than once.



# FINITO

- Brian Blood - [brian@macserve.net](mailto:brian@macserve.net)
- Blog - <http://www.networkjack.info/blog/>
- <http://joind.in/2973>
- Biz:
  - MacServe.net - hosting / development / server colo
  - DFWDDataVault.com - Smart and Easy Backup
  - AHDS - Health Care Data and Connectivity