Lessons learned in a D7 upgrade of origamiusa.org

Robert J. Lang
Managing editor, OrigamiUSA.org
Welcome to OrigamiUSA
We are the American national society devoted to origami, the art of paperfolding.

Origami is for anyone, anywhere, any time!

Our website is the place to find easy diagrams to fold from, shop from our broad selection of origami books, paper, and downloadable diagrams, and find events from our calendar such as our fantastic Conventions that bring folders together from all over the world. Find an origami group near you or read our online magazine for more diagrams and great articles. Join us and you receive our print magazine. Experience the joy of paperfolding!

Our Facebook
Volunteers needed for the convention Please remember that the Source as well as...
Posted: Jun 18 2013 - 6:30pm
Yay - the last expected box of new items arrived today (nothing like calling it...
Posted: Jun 18 2013 - 6:28pm

Diagram Downloads

Current News
• Convention 2013: June 21-24, 2013, open for registration
Site structural overview

- Drupal 6
  - Custom theme (derived from Basic)
- CiviCRM
  - membership, donations, event registrations
- Ubercart
  - online store selling books and paper
- Mailman mailing lists
  - User Mailman Register + custom modules
CiviCRM membership status used to determine display of Drupal blocks and display of individual node and View content.

Diagrams: Pureland Modular Rose

You must be a current member of OrigamiUSA to access this article. Join or renew.

by Jason Ku

Diagrams for a modular pureland rose made from 1:2 rectangles.
Customizations

- 67 assorted contrib modules
- 24 custom-written modules
  - 8 customizations to Ubercart
  - 2 customizations to CiviCRM
  - 14 others implement organization-specific behaviors
Ubercart Customizations

- Buyer Comments - you can comment on a product if you’ve bought it
- Seller Downloads - “iTunes for file downloads”
- PDF File Marker - watermark sold file downloads with user credentials
- Product Context - catalog shows if you’ve previously bought a product
- Tax NJ - State-dependent NJ sales tax
- Tax NY - Locality-dependent NY sales tax
- USPS Packager - automatically packs products and gets cheapest multi-box rates from USPS
- OUSA - miscellaneous (special reports for store manager, minor overrides/alters)
Usage

- Light traffic: ~1000 visitors/day
- Store e-commerce: 3–5 orders/day
- CiviCRM transactions: 5–10 memberships, registrations, donations/day
Downtime tolerance

- No loss of data or transactions permitted
- Decoherence time could be minutes
- Short outages tolerable with advance notice and explanatory page
- 1–2 hours max
Desired upgrade process

• Take site offline
• Do the upgrade
• Bring site back
• All within 1-2 hours
• AND IT MUST BE FULLY FUNCTIONAL AS SOON AS IT’S BACK.
Develop locally, push to SVN
Stage to sandbox (via shell script)
Deploy to production (via shell script)
Constraints

• Minimize offline time
• High degree of assurance that upgrade will “work”
  • Rehearse, rehearse, rehearse
• ==> run the entire upgrade via script
Why use a script?

• Surprises happen
• Multiple rehearsals possible
• Guaranteed repeatability of sequence of actions
• Upgrade process can be developed over time
• Upgrade can be broken down into tasks
Script toolkit

• bash

• rsync, chmod, chown...

• drush (Drupal Shell, http://drush.org)

• drush commands (predefined)

• drush scripts (execute PHP)
Global options (see `drush topic core-global-options` for the full list):
- `-d`, `--debug` Display even more information, including internal messages.
- `-h`, `--help` This help system.
- `-i`, `--interactive` Force interactive mode for commands run on multiple targets (e.g. `drush @site1,site2 cc --ia`).
- `-n`, `--no` Assume 'no' as answer to all prompts.
- `--php=/path/to/file` The absolute path to your PHP interpreter, if not 'php' in the path.
- `-p`, `--pipe` Emit a compact representation of the command for scripting.
- `-r <path>`, `--root=<path>` Drupal root directory to use (default: current directory).
- `-s`, `--simulate` Simulate all relevant actions (don't actually change the system).
- `--uri=http://example.com:8888`,
- `--url=http://example.com:8888`
- `--version` Show drush version.
- `--yes` Assume 'yes' as answer to all prompts.

Core drush commands: (core)
- `archive-dump (ard)` Backup your code, files, and database into a single file.
- `archive-backup (arb)`
- `archive-restore (arr)` Expand a site archive into a Drupal web site.
- `cache-clear (cc)` Clear a specific cache, or all Drupal caches.
- `cache-get (cg)` Fetch a cached object and display it.
- `cache-set (cs)` Cache an object expressed in JSON or var_export() format.
- `core-config (conf, config)` Edit drushrc, site alias, and Drupal settings.php files.
- `core-cron (cron)` Run all cron hooks in all active modules for specified site.
- `core-execute (exec, execute)` Execute a shell command. Usually used with a site alias.
- `core-quick-drupal (qd)` Download, install, serve and login to Drupal with minimal configuration and dependencies.
- `core-requirements (status-report, rq)` Provides information about things that may be wrong in your Drupal installation, if any.
- `core-rsync (rsync)` Rsync the Drupal tree to/from another server using ssh.
- `core-status (status, st)` Provides a birds-eye view of the current Drupal installation, if any.
- `core-topic (topic)` Read detailed documentation on a given topic.
- `drupal-directory (dd)` Return path to a given module/theme directory.
- `help` Print this help message. See `drush help` for more options.
- `image-flush` Flush all derived images for a given style.
- `php-eval (eval, ev)` Evaluate arbitrary php code after bootstrapping Drupal (if available).
- `php-script (scr)` Run php script(s).
- `queue-list` Returns a list of all defined queues
- `queue-run` Run a specific queue by name
- `search-index` Index the remaining search items without wiping the index.
- `search-reindex` Force the search index to be rebuilt.
What can you do with drush?

* Darn near anything:
  * Enable/disable modules
  * Set configuration variables
  * Move blocks around, turn on/off
  * Override content types, Views, other settings
  * Call any Drupal API function
  * Make direct changes to the db
Why not just bash?

- drush bootstraps Drupal for you
- drush exposes a hierarchy of access
  - high-level: drush commands (e.g., enable a module)
  - medium level: Drupal APIs (e.g., delete field instance)
  - low-level: direct PHP and MySQL queries
- Use the highest-level command that does the job
  - minimizes unintended side effects/broken dependencies
Philosophy

• Everything you might do from the UI should be accomplished in code and captured in the script.
Upgrade in 5 easy steps

- According to Drupal upgrade docs:
  - Take D6 site offline
  - Disable non-core modules
  - Deploy D7 code
  - drush updatedb (= update.php)
  - Bring D7 site back online. Done!
Things break!

- CCK fields no longer supported, replaced by Fields
- Content types don’t display the same
- Many existing Views break
- Our custom modules don’t have D7 versions (yet)
- Neither does our custom theme
- Contrib modules may not exist in D7
Divide and conquer
Things break!

- CCK fields no longer supported
- Content types don’t display the same
- Many existing Views break
- Our custom modules don’t have D7 versions (yet)
- Neither does our custom theme
- Contrib modules may not exist in D7
CCK Fields

- CCK goes away in D7; replaced by Field API in Drupal Core.
- The Content Migration (content_migrate) module migrates all your old CCK fields to new Field API fields for you.
- And it supports drush:
  - drush content-migrate-fields
Things break!

- CCK fields no longer supported
- Content types don’t display the same
- Many existing Views break
- Our custom modules don’t have D7 versions (yet)
- Neither does our custom theme
- Contrib modules may not exist in D7
Content type display

- Content migration of fields loses display settings on CCK fields
- From UI, “edit content types”; restore field order, display settings, formatters
- But how to capture in a script?
Features Module

- Features module moves configuration data from UI+db into code (and vice-versa)
- An individual “Feature” is a Drupal module
- Similar to, e.g., hook_views_default.inc in D7
Features actions

UI+db

create
revert
override
update
codebase
Features supports drush

All commands in features: (features)

features-add (fa) Add a component to a feature module. (DEPRECATED: use features-export)
features-components List features components.
(features)
features-diff (fd) Show the difference between the default and overridden state of a feature.
features-export (fe) Export a feature from your site into a module.
features-list (fl, features)
features-revert (fr) Revert a feature module on your site.
features-revert-all (fr-all, fra)
features-update (fu) Update a feature module on your site.
features-update-all (fu-all, fua)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Signature</th>
<th>State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Content Type Board Notes</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Calendar Event</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Diagrams PDF</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Download Submission</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Gallery Photo</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Magazine Article</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Magazine Issue</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type News Story</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Origami Group</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type OrigamiUSA Publication</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Page</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Publication Submission</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type Special Sessions</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
<tr>
<td>Feature Content Type The Paper Issue</td>
<td>Unavailable</td>
<td>Default</td>
<td>Recreate</td>
</tr>
</tbody>
</table>
Features in upgrade strategy

• Perform the “vanilla” upgrade
• Tweak the db via UI to fix breakages
• Capture in Feature
• Add Feature to codebase
• Add enabling to upgrade script:
  • drush pm-enable feature_my_feature
De-featurizing

• Features model: “once in code, forever in code”
• Development in progress: “features-import” to put everything back into the db
• This works, too:
  • [https://drupal.org/node/663136#comment-7861781](https://drupal.org/node/663136#comment-7861781)
• D8’s Configuration Management is a lot like this
Things break!

- CCK fields no longer supported
- Content types don’t display the same
- Many existing Views break
- Our custom modules don’t have D7 versions (yet)
- Neither does our custom theme
- Contrib modules may not exist in D7
Features again

- You can build a feature for any view.
- Almost...
Features limitations

- Features moves config data from the db into code

- But what happens if it’s ALREADY in code?

  (e.g., hook_views_defaults.inc)
Badness

- Blank feature (nothing saved), or
- PHP errors.
Views in code

• Ubercart 3 (D7) provides standard displays as Views defined in code

• Ubercart defines its product content types in code

• How do we tweak them if Features don’t work for stuff already in code?
Features-like “clobber”

- drush script to override content type or View
- Use CTools Export to get code description of config data, then paste into script
<?php
// This script overrides the uc_catalog_terms View. Implemented via script rather than Feature.
// because the View is in code.

// Purpose of this override is to child term links to the catalog grid page.
//
// drush scr /scripts/drush/d7/override_view_uc_catalog_terms.php

// Modify the view in the UI, export it, then paste in the exported definition here.
$view = new View();
$view->name = 'uc_catalog_terms';
$view->description = 'Catalog categories. Provides grid view on main catalog page.';
...

// To wrap up, we give new newly created view the vid of the one we're replacing, then save it.
// See sites/all/modules/views/include/admin.inc, function views_ui_import_validate()

$test = views_get_view($view->name);
if (!$test) {
    exit('there is no view with name ' . $view->name . "\n") ;
}
$view->vid = $test->vid;
views_save_view($view);
?>
Content types

- Content types have no UI “export” function
- So, we write our own
Content type override

- Tweak content type in the UI
- Run a script, execute `write_field_info()` to spew content type as JSON output
- Copy output
- Paste into script as argument of `read_field_info()`
- Add to master upgrade script:
  - `drush scr my_override_script.php`
/** *
 * Write a JSON string of the field information and bundle settings for the given entity type and bundle. This is how we capture a group of settings.
 */

function write_field_info($entity_type, $bundle_name) {
 $instances = field_info_instances($entity_type, $bundle_name);
 $settings = field_bundle_settings($entity_type, $bundle_name, NULL);
 $data = array(
 'entity_type' => $entity_type,
 'bundle_name' => $bundle_name,
 'instances' => $instances,
 'settings' => $settings,
);
 print("\n\n");
 print(drupal_json_encode($data));
 print("\n\n");
}

/** *
 * Read in a JSON string of the field information and bundle settings for the given entity type and bundle and overwrite the existing settings.
 */

function read_field_info($data) {
 if (!$data) {
   return;
 }

$data = drupal_json_decode($data);
$entity_type = $data['entity_type'];
$bundle_name = $data['bundle_name'];
if (!$entity_type || !$bundle_name) {
   print("Bad data in read_field_info()\n");
   return;
}

// Update each of the field instances
foreach ($data['instances'] as $instance) {
   field_update_instance($instance);
}

// Update the bundle settings
field_bundle_settings($entity_type, $bundle_name, $data['settings']);
print("Updated entity " . $entity_type . ", bundle " . $bundle_name . "\n");

// Wipe out the residual file attachment field, if present
$instances = field_read_instance($entity_type, 'upload', $bundle_name);
if ($instance) {
   field_delete_instance($instance);
}
}
Minor content type tweaks can be done directly in code

```php
<?php
// This script overrides the display format for the FAQ to suppress listing of
// category terms. Need to do it in a script instead of a content type Feature
// because the content type is already in code so Features can't touch it.

// drush scr /scripts/drush/d7/override_faq_display.php

$instance = field_info_instance('node', 'taxonomy_vocabulary_7', 'faq');
//print_r($instance);
$instance['display']['default']['type'] = 'hidden';
$instance['display']['teaser']['type'] = 'hidden';
field_update_instance($instance);
?>
```
Things break!

- CCK fields no longer supported
- Content types don’t display the same
- Many existing Views break
- **Our custom modules don’t have D7 versions (yet)**
- Neither does our custom theme
- Contrib modules may not exist in D7
Custom modules

- Written over several years; due for a cleanup anyhow
- Good practice for D7 porting, because I already know what they’re supposed to do
Useful tools for porting

- Drupal 7 APIs (of course)
- “Converting 6.x to 7.x modules” https://drupal.org/update/modules/6/7
- Browse existing Core and Contrib D7 modules
- Biggest changes I found:
  - New database API
  - hook splitting (hook_node --> hook_node_save, etc.)
  - Return render arrays, not HTML
- Devel module and Coder are your friend
Things break!

- CCK fields no longer supported
- Content types don’t display the same
- Many existing Views break
- Our custom modules don’t have D7 versions (yet)
- Neither does our custom theme
- Contrib modules may not exist in D7
Origami theme

• custom theme, based on Basic (D6)

• Basic (D7) exists, but many things change from D6, e.g.:

  • structure of page.tpl.php

  • names of blocks (“left” --> “sidebar_first”)

• In D6, most site-specific CSS is dumped into theme style.css & template.php
Theme porting

- Started with Basic (D7)
- Copied CSS from origami theme with modifications as needed
- Moved CSS into custom modules where possible to pare down style.css in theme
D6 & D7
Things break!

- CCK fields no longer supported
- Content types don’t display the same
- Many existing Views break
- Our custom modules don’t have D7 versions (yet)
- Neither does our custom theme
- Contrib modules may not exist in D7
Contrib Modules

• Aye, there’s the rub.
Many contribs support D7

- ...and provide a migration path via drush updatedb (update.php)
- CiviCRM
- Ubercart
- ...and many others.
Problematic contribs

- Taxonomy Super Select
  - turned large hierarchical taxonomies into tractable nested boxed lists

- Private Upload
  - Access-controlled file attachments
  - private file storage is in core in D7, but no migration path exists for this module

- Mailman Manager, User Mailman Register
  - provides Drupal interface to Mailman mailing lists
Taxonomy Super Select

- Used for the Ubercart catalog
(D6) Taxonomy Super Select

turns this into this
(D7) Term Reference Tree

turns this into this
But with some CSS...

turns this into this
Private Upload

- Private Upload module allows user to choose “public” or “private” (access-controlled) storage for file attachments
- D7 changes file storage model to offer both public and private file storage in Core, BUT
- All D6 files get assigned to “public” storage by default during upgrade
- The URLs for private files are different in D7 than they were in D6 with Private Upload module
  - this breaks pages with embedded <img/> files.
Private file use cases

• We have members-only content
  • images in members-only articles
  • publication submissions from users
• members-only organizational documents (elections, financials)
Strategy

• Give content types a new private file field if needed (capture in Feature)

• Go through all nodes and move formerly private files to new private file field

• Go through all node bodies in db, find ‘src="old_url"’ and replace with ‘src="new_url"’, etc.

• drush script for each content type (~200 LOC)
User Mailman Register

- Mailman Manager and User Mailman Register work together to provide a Drupal interface for Mailman mailing lists
- No visible activity toward a D7 version
- No other D7 modules do anything with Mailman
- Our organization is heavily tied to Mailman mailing lists
If all else fails...

• By this time:

• I’d spent six months porting our theme and my own modules to D7
So I ported it.

• I’m now an author and co-maintainer of the D7 User Mailman Register project: http://drupal.org/project/user_mailman_register.
Put it all together!

- Here’s our master script.
- Each sub-script leaves the site in a (relatively) stable state.

```bash
#!/bin/sh
#
# This script does a full run of the upgrade process on sandbox or production.
#
# /scripts/d7-all.sh
#
# Stop the script if any error happens.
set -e
#
# Prepare the D6 version for the upgrade.
# /scripts/d7-prepare.sh
#
# Switch the codebase to D7 and run upgradedb.
# /scripts/d7-svn_deploy.sh
#
# Run all the update functions to update the db.
# /scripts/d7-updatedb.sh
#
# Enable core and contrib modules and migrate the data.
# /scripts/d7-migrate.sh
#
# Enable CiviCRM, Ubercart, OrigamiUSA modules, and Features.
# /scripts/d7-enable_more.sh
#
# Tweaking, configuration, and overrides.
# /scripts/d7-tweak.sh
#
# Uninstall unused modules and drop obsolete db tables.
# /scripts/d7-uninstall.sh
#
# LOCAL INSTALLATION ONLY - configure for local development.
# /scripts/d7-devel.sh
#
echo "Done with d7-all.sh"
```
Prepare

# Take site offline-
echo "Taking site offline..."
sudo drush vsat --exact maintenance_mode

# Special steps to convert CCK Content Permissions to Field Permissions-
sudo drush pm-enable field_permissions --yes
sudo drush pm-disable content_permissions --yes
sudo drush pm-uninstall content_permissions --yes

# Enable Garland, make it the default and admin theme-
sudo drush pm-enable --yes garland
sudo drush vsat theme_default garland
sudo drush vsat admin_theme garland

# Disable non-Core - required, non-Core - optional modules-
sudo drush pm-disable --yes "drush pm-list --type=module --package="Access control,Administration,CCK,Chaos tool suite,CiviCRM,Custom Breadcrumbs,Database,Date/Time,Development,Features,Feeds,File management,ImageCache,Input filters,Javascript timer,Mailman,Beaver,Meta tags,OpenID,USA - other,Other,Pages,Rules,Spam control,Testing,Ubercart - core,Ubercart - core (optional),Ubercart - extra,Ubercart - fulfillment,Ubercart - payment,Ubercart - views,User interface,Util,Views,Webform" --status=enabled --pipe"

# Also disable the core PHP module, because it causes conflicts later.-
sudo drush pm-disable php --yes

# Uninstall D6 modules we won't need, note, order matters.-
sudo drush pm-uninstall taxonomy super_select --yes
sudo drush pm-uninstall uc_views --yes

# Tweaks to restore taxonomy information so that Taxonomy updating goes well for FAQ and Ubercart taxonomy-
sudo drush sql-query "INSERT INTO vocabulary_node_types VALUES (?, 'faq')"
sudo drush sql-query "INSERT INTO vocabulary_node_types VALUES (24, 'product')"
sudo drush sql-query "INSERT INTO vocabulary_node_types VALUES (24, 'nortax')"
sudo drush sql-query "INSERT INTO vocabulary_node_types VALUES (24, 'product_download')"
sudo drush sql-query "INSERT INTO vocabulary_node_types VALUES (24, 'productkit')"

# At this point, we still have a minimally functional D6 site (in maintenance mode)-
# and we're ready to install the D7 files (d7_svn_switch.sh) and then start-
# enabling D7 modules (d7_config.sh)-
echo "Remaining enabled modules!"-
sudo drush pm-list --status=enabled
Deploy from SVN

• ...

...
Run updatedb

# Run update.php -- need to do this before anything else will function (even drush cc).
sudo drush updatedb --yes

# This directory is created as debris during upgrade; we can remove it.
sudo rm -dfR sites/default/files

# Clear drush's own cache since we've changed the codebase so drush commands have changed.
sudo drush cache-clear drush

# Bring site back online
echo "Bringing site back online..."
sudo drush vdel -y --exact maintenance_mode
Enable and migrate

- Re-enable core modules.
  sudo drush pm-enable php --yes

- Enable D7 contrib modules. Some of these will already have been enabled due to dependencies,
  but there's no harm in trying and this gives a complete listing.
  echo "Enabling D7 contrib modules..."
  sudo drush pm-enable views --yes
  sudo drush pm-enable views_content --yes
  sudo drush pm-enable admin_menu --yes
  sudo drush pm-enable advanced_help --yes
  sudo drush pm-enable captcha --yes
  sudo drush pm-enable image_captcha --yes
  sudo drush pm-enable content_access --yes
  sudo drush pm-enable ctools --yes

- ...

- snip

- ...
  sudo drush pm-enable views_rss --yes
  sudo drush pm-enable webform --yes
  sudo drush pm-enable wysiwyg --yes
  sudo drush pm-enable link --yes
  sudo drush pm-enable views_ui --yes
  sudo drush pm-enable features --yes
  sudo drush pm-enable term_reference_tree --yes
  sudo drush pm-enable menu_breadcrumb --yes
  sudo drush pm-enable contextual --yes

- Run update.php again, backup_migrate will do some further updating.
  sudo drush updatedb --yes

- clear cache to make content-migrate-fields available
  sudo drush cc all

- Migrate content fields using the default settings.
  echo "Starting field migration..."
  sudo drush content-migrate-fields --yes

-
Enable modules & Features

- sudo drush pm-enable feature_image_styles_custom --yes
- sudo drush pm-enable feature_text_formats --yes
- sudo drush pm-enable feature_feeds_tamper_origami_usa_facebook --yes
- sudo drush pm-enable feature_content_type_board_notes --yes
- sudo drush pm-enable feature_content_type_calendar_event --yes
- sudo drush pm-enable feature_content_type_diagrams_pdf --yes
- sudo drush pm-enable feature_content_type_downloads_submission --yes
- sudo drush pm-enable feature_content_type_gallery_photo --yes
- sudo drush pm-enable feature_content_type_magazine_article --yes
- sudo drush pm-enable feature_content_type_magazine_issue --yes
- sudo drush pm-enable feature_content_type_news_story --yes
- sudo drush pm-enable feature_content_type_origami_group --yes
- sudo drush pm-enable feature_content_type_origami_usa_publication --yes
- sudo drush pm-enable feature_content_type_page --yes
- sudo drush pm-enable feature_content_type_publication_submission --yes
- sudo drush pm-enable feature_content_type_special_sessions --yes
- sudo drush pm-enable feature_content_type_the_paper_issue --yes
- sudo drush pm-enable feature_rules_comments --yes
- sudo drush pm-enable feature_views_art_gallery --yes
- sudo drush pm-enable feature_views_board_notes --yes
- sudo drush pm-enable feature_views_calendar_events --yes
- sudo drush pm-enable feature_views_civicrm_recurring --yes
- sudo drush pm-enable feature_views_convention_classes --yes
- sudo drush pm-enable feature_views_diagrams --yes
- sudo drush pm-enable feature_views_facebook_feed --yes
- sudo drush pm-enable feature_views_galleru --yes
- sudo drush pm-enable feature_views_header_barner --yes
- sudo drush pm-enable feature_views_holiday_tree --yes
- sudo drush pm-enable feature_views_hot_news --yes
- sudo drush pm-enable feature_views_lending_library --yes
- sudo drush pm-enable feature_views_local_groups --yes
- sudo drush pm-enable feature_views_magazine --yes
- sudo drush pm-enable feature_views_usapubs --yes
- sudo drush pm-enable feature_views_pubsub --yes
- sudo drush pm-enable feature_views_special_sessions --yes
- sudo drush pm-enable feature_views_the_paper --yes
- sudo drush pm-enable feature_views_uc_catalog_header --yes
- sudo drush pm-enable feature_views_uc_downloads_newest --yes
- sudo drush pm-enable feature_views_uc_orders_details --yes
- sudo drush pm-enable feature_views_uc_products_clearance --yes
- sudo drush pm-enable feature_views_uc_products_details --yes
- sudo drush pm-enable feature_views_uc_products_labels --yes
- sudo drush pm-enable feature_views_uc_products_pages --yes
- sudo drush pm-enable feature_views_uc_products_unpub --yes
- sudo drush pm-enable feature_views_users_by_role --yes
- sudo drush pm-enable feature_views_users_special --yes
- sudo drush pm-enable feature_views_users_all --yes
- sudo drush pm-enable feature_views_webform_results --yes
- sudo drush features-revert-all --yes
Tweak

- Run drush scripts to override non-Featureable views and content types
- Move blocks around
- Set lots of configuration parameters
- Move private files around and make associated changes
- ...and more (230 LOC in shell script + 18 separate drush scripts)
Uninstall

• Schema module reveals unused db tables (e.g., leftovers from Views 1, obsolete CCK tables)

• drush sql-query “DROP TABLE ...” to clear them out
Putting it together

- Many partial runs on local system while developing scripts and process
- Rehearsal run on our sandbox
  - live server similar to production
  - under httpd password protection
- ~2 weeks of user testing on sandbox
  - Users find “wallpaper”
Pull the trigger

- Upgrade performed on live system on 07/21/2013
- Script ran for ~90 minutes
- Back online and running in <2 hours
Our website is the place to find easy diagrams to fold from, shop from our broad selection of origami books, paper, and downloadable diagrams, and find events from our calendar such as our fantastic Conventions that bring folders together from all over the world. Find an origami group near you or read our online magazine for more diagrams and great articles. Join us and you receive our print magazine. Experience the joy of paperfolding!

Our Facebook

- Origami as Fine Art http://origamifineart.com/
- Postcard June 19, 2013 - 11:31am
  Volunteers needed for the convention. Please remember that the Source as well as...
- Postcard June 18, 2013 - 6:30pm
  Yay - the last expected box of new items arrived today 
  (nothing like calling it...)
- Postcard June 18, 2013 - 6:28pm
  "International Teachers at OUSA Convention"
  We're looking for something new and...
- Postcard June 16, 2013 - 2:22pm
  A little addition to our convention activities, as follows: 
  "Art & Architecture at OUSA Convention"
- Postcard June 15, 2013 - 1:50pm
  Origami Butterflies Mini Kit. Contains additional models 
  not found in the Origami...
- Postcard June 15, 2013 - 1:43pm
  "Whaddaya attending the origami convention in NYC? 
  Follow us on Facebook"

Diagram Downloads

- Romantic Origami [pdf]
  Regular price: $5.95, Member price: $5.36
- The Many Variations on a Star [pdf]
  Regular price: $2.00, Member price: $1.80
- 8 Pointed Star [pdf]
  Regular price: $1.00, Member price: $0.90
- Manta ray [pdf]
  Regular price: $1.00, Member price: $0.90
- Dollar Christmas Tree On A Stand (PDF) [pdf]
  Regular price: $2.00, Member price: $1.80
- Christmas Present - Santa Claus [pdf]
  Regular price: $3.00, Member price: $2.70
- One Dozen Folds [pdf]
  Regular price: $5.95, Member price: $5.36
- Clover Leaf [pdf]
  Regular price: $1.89, Member price: $1.70

The newest downloads in The Source.

Current News

- Convention 2013: June 21-24, 2013, open for registration
- Spring Special Folding Fun Sessions are open for Registration
- Letter from the President of OrigamiUSA
- PCOC 2013 in Albuquerque, NM: October 3-7, 2013
- Free Annual Collections for Schools, Libraries, and Local Origami Groups
The result

- Numerous small glitches showed up once the public started pounding on the site
- Nothing major, all easily/quickly fixable
- Closely monitor dblog for the first week
Team credits

- Me (managing editor, Drupal programming)
- Anne LaVin, Jim Weir (sysadmins)
- Trevor James (original site design, Drupal consultant)
Final thoughts

• Script-based upgrade is ABSOLUTELY the way to go

• Upgrading a site is a great way to get to know Drupal innards