

a paradigm for reusable drupal features

Ian Ward
Robert Soden
Young Hahn

development seed



- 1 In theory: the problem + solution
- 2 In practice: feature building demo
- 3 The future: what comes next

Screenshot of first blog:



Blog

6.10

Enables keeping easily and regularly updated user web pages or blogs.



Blogs

[Create new blog entry.](#)

My Second Post

Thu, 02/26/2009 - 11:14 — [admin](#)

Vestibulum sapien. Donec diam justo, egestas et; posuere quis, tempus et; turpis! Suspendisse lacinia. Curabitur fermentum quam faucibus elit. Suspendisse potenti. Etiam mattis enim quis sem. Vivamus a leo. Vivamus eu quam sed dui hendrerit consectetur. Fusce rhoncus fringilla pede. Morbi porta consequat nisl. Maecenas lobortis lacinia tortor? Quisque euismod nunc. Praesent gravida cursus risus. Maecenas nec justo vitae sapien consectetur elementum. Aenean in lorem. Quisque ipsum dolor, adipiscing non, ultricies vel, cursus at, purus!

[admin's blog](#) [Add new comment](#) [Read more](#)

Hello, World

Thu, 02/26/2009 - 11:10 — [admin](#)

Etiam vitae ligula luctus odio eleifend egestas. Curabitur nisl nibh; sodales eu, venenatis gravida, luctus ac, nibh. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vestibulum odio in risus. In hac habitasse platea dictumst. Sed odio. Etiam non sapien vitae metus lacinia consequat. Ut eu ligula vel elit porttitor semper. Pellentesque blandit leo eget massa condimentum tincidunt? In pellentesque. Morbi blandit condimentum tellus.

[admin's blog](#) [1 comment](#) [Read more](#)



A blog is a "feature"

and so is:

- a pressroom,
- an image gallery,
- a publication library,
- etc. etc.

Let's make a pressroom

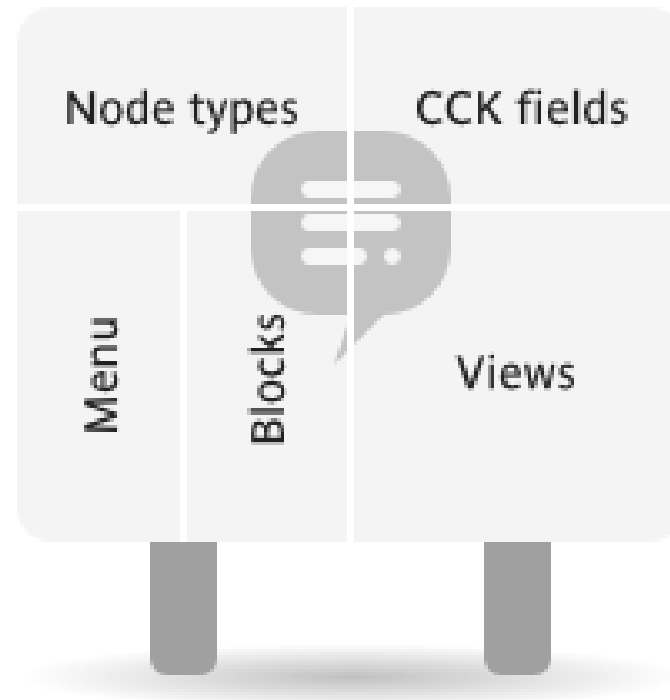
- 1 create press release content type
- 2 create a subtitle field
- 3 create imagecache preset and imagefield
- 4 create a page view
- 5 press contact block
- 6 related press releases views block
- 7 create a vocabulary
- 8 create tag cloud views block
- 9 ...

3 problems:

- 1 slow process
- 2 not reusable
- 3 disconnected pieces

Context

this is the pressroom.



Composition of a pressroom feature

From user interface to code.

Drupal feature modules are born

Anatomy of a feature

content type	scripted	node
fields	scripted	CCK
taxonomy	scripted	taxonomy
menus	scripted	menu
imagecache presets	exportable	imagecache
block visibility	exportable	context_ui
block views	exportable	views
page views	exportable	views
comp. relations	exportable	context_ui

2003 to 2009...

**Blog**

6.10

Enables keeping easily and regularly updated user web pages or blogs.

Features	Status	Description	
 Blog	Enabled 	Blog Landing Page.	Customize



[who we are](#)
[what we do](#)

The Problem

Policy Research to extend antibiotic effectiveness

Features

[Basic setup](#)
[Features](#)

Features

Features	Status	Description
+ The Problem Book	Enabled	
+ Solutions Book	Enabled	
+ Resource Library	Enabled	Resource Lib
+ Blog	Enabled	Blog Landin
+ Press Room	Enabled	



[Home](#)
[My Account](#)
[Create](#)
[Logout](#)

Members Settings

Development Seed

[View](#)
[Map](#)
[Edit](#)
[Spaces](#)

Features

Features

Features	Status	Description
+ Blog	Private	A blog for team comm
+ Casetracker	Private	Provides an issue and
+ Documents	Private	A document section fo
+ Calendar	Private	Provides a team calen



the global resource in.itiative.org

in unity lies strength

[home](#)
[about](#)
[team](#)
[blog](#)

Features

[Basic setup](#)
[Features](#)

Features

Features	Status	Description
+ Admin	Enabled	Tools for adn
+ Blog	Enabled	Blog Landin
+ Press Room	Enabled	Pressroom s
+ BigDaddy	Enabled	BigDaddy.
+ About Book	Enabled	
+ Press Resources Book	Enabled	



developmentSEED

[about](#)
[solutions](#)

Features

Features

Features	Status	Description
+ Admin	Enabled	Tools for administering content.
+ Home	Enabled	DS homepage display.
+ About	Enabled	A document section for maintaining
+ Solutions	Enabled	DS solutions section.
+ Clients	Enabled	DS clients section.
+ Team	Enabled	Provides a listing of DS team memb

Demo

Why do we want our
site building in code?

- 1 Code is reusable
- 2 Code is distributable
- 3 Code can be kept under version control

How do we get our
site building into code?

Scripts

An example script:

- 1 Create a pressroom content type
- 2 Add CCK fields to it
- 3 Create a pressroom menu item
- 4 Create “contact info” and “archives” subnav items in the menu

- 1 Is there already a pressroom content type?
 - > Create a pressroom content type
- 2 Does the pressroom content type have any fields I can use?
 - > Add CCK fields to it
- 3 Is there already a pressroom menu item?
 - > Create a pressroom menu item
- 4 What subnav items are in the menu?
 - > Create “contact info”
 - > Create “archives”

- 1 scripts are a pain to debug
- 2 scripts are a pain to maintain
- 3 scripts are not easily extensible

Exportables

Views

[List](#)[Add](#)[Import](#)[Tools](#)

If you install the advanced help module from http://drupal.org/project/advanced_help, Views will provide more and better help. [Hide this message.](#)

Not sure what to do? Try the "Install the advanced help module for the getting started" page.

Storage:

<All> ▾

Type:

<All> ▾

Tag:

<All> ▾

Displays:

<All> ▾

Sort by:

Name ▾

Order:

Up ▾

Normal Node view: **blog** (demo)[Edit](#) | [Export](#) | [Clone](#) | [Delete](#)

My blog

Default Node view: **frontpage** (default)[Edit](#) | [Export](#) | [Clone](#) | [Disable](#)Path: [frontpage](#)*Feed, Page*

Emulates the default Drupal front page; you may set the default home page path to this view to make it your front page.

Overridden Node view: **taxonomy_term** (default)[Edit](#) | [Export](#) | [Clone](#) | [Revert](#)

Path: taxonomy/term/%

Feed, Page

A view to emulate Drupal core's handling of taxonomy/term; it also emulates Views 1's handling by having two possible feeds.

Default Node view: **archive** (default)[Enable](#)

Exportables abstract data storage
so that your module can be
indifferent to whether an item is in
the database, code, or both.

- 1 exportables provide definitions, not scripts
- 2 exportables are easy to maintain
- 3 exportables are overridable and extensible

Who's got exportables?

views

views

panels

panels

context_ui

context definitions

imagecache

imagecache presets

Who doesn't?

node / CCK

content types + fields

taxonomy

vocabularies

filter

input formats

menu

menus

triggers

triggers/actions

your module ?

???

Recipe for exportables

1 string identifiers

good: `taxonomy_term` (view), `thumbnail` (imagecache preset)

bad: `5` (vocabulary)

2 API functions for collecting all items

e.g. `node_get_types()`, `imagecache_preset()`

3 Structured array or object to represent your item

```
function mymodule_items() {  
    $items = array();  
  
    // Load all items from the database  
    $result = db_query();  
    while($row = db_fetch_array($result)) { ... }  
  
    return $items;  
}
```

```
function mymodule_items() {  
  
    // Collect default items  
    $items = module_invoke_all('mymodule_default_items');  
    drupal_alter('mymodule_default_items', $items);  
  
    // Load all items from the database  
    $result = db_query();  
    while($row = db_fetch_array($result)) {  
        ...  
        // Set storage status  
        if (!empty($items[$row->id])) {  
            $row->storage = MYMODULE_OVERRIDDEN;  
            $items[$row->id] = $row;  
        }  
    }  
  
    return $items;  
}
```



ImageCache

[List](#)[Add new preset](#)

Manage ImageCache presets.

Preset Name	Actions		
large	Edit	Delete	Flush
small	Edit	Delete	Flush
thumbnail	Edit	Delete	Flush



ImageCache

[List](#)[Add new preset](#)

Manage ImageCache presets.

Preset Name	Storage	Actions			
large	Normal	Edit	Delete	Flush	Export
small	Normal	Edit	Delete	Flush	Export
thumbnail	Normal	Edit	Delete	Flush	Export

```
function mymodule_item_export($item) {  
    $form = array(  
        '#title' => t('My export'),  
        '#type' => 'textarea',  
    );  
    $export = '$items[] = '.var_export($item, TRUE);  
    $form['#default_value'] = $export;  
    return $form;  
}
```

Export preset: thumbnail

Place the following snippet in your module as part of `hook_imagecache_default_presets()`.

```
$presets = array();
$presets['thumbnail'] = array (
  'presetname' => 'thumbnail',
  'actions' =>
  array (
    0 =>
    array (
      'weight' => '0',
      'module' => 'imagecache',
      'action' => 'imagecache_scale_and_crop',
      'data' =>
      array (
        'width' => '60',
        'height' => '60',
      ),
    ),
  ),
);
```

featuremaker.module ?

The future of Drupal features?

Imagine a module that would automatically package any of your site's features into a module for you to download and share.

If you are a developer, add exportables to your module.

If you're not, request exportables as a feature on your module's issue queue.

now all we need is...

`do_my_issue_queue.module`

Thank you

<http://www.developmentseed.org/tags/spaces>

<http://svn3.cvsdude.com/devseed/sandbox/drupal-6/myblog>

Twitter #spaces

Context <http://drupal.org/project/context>

Spaces <http://drupal.org/project/spaces>

Chaos tools (interesting code for developers)

<http://drupal.org/project/ctools>

Flickr CC

<http://flickr.com/photos/usefulguy/226362739>

<http://flickr.com/photos/usefulguy/226362796>