# How, When and Why to Patch a Module

May 13, 2015



Wednesday, May 13, 15

Good Afternoon! [Introduction and overview.]

# What's a Patch?



Wednesday, May 13, 15

When you are working with a code platform like Drupal, there may come a time when you need to make a change to some code that you didn't write, or don't control.

Patches are how we do that. They are a universal standard in the software world - not just in

the web world, but video games, enterprise business applications, and operating systems all use patches to make changes.

## Patch File



秒

Wednesday, May 13, 15

At their heart, patches are small files that contains a list of all the differences between the existing code, and the code you want to run.

They are "applied" to the existing code, and make permanent changes \*to\* that code.

#### This is an example of a patch file. Let's take a closer look...

```
diff --git a/patch_demo.module b/patch_demo.module
index e835fbb..7ff9565 100644
                               Patch File
--- a/patch_demo.module
+++ b/patch_demo.module
@@ -125,7 +125,7 @@ function patch_demo_list() {
      $rows[] = $entry_row;
    // Make a table for them.
    $header = array(t('Id'), t('Name'), t('Surname'), t('Edit'));
  $header = array(t('Id'), t('Name'), t('Surname'), t('Job Title'), t('Edit'));
     sutput .= tneme( table , array( neader => $neader, rows => $rows));
  else {
@@ -165,6 +165,19 @@ function patch_demo_user_form($form, &$form_state, $pid = NULL)
     '#default_value' => !empty($default_entry) ? $default_entry[0]->surname : "",
  );
  $options = array(
+
 'developer' => 'Developer',
+
  'project_manager' => 'Project Manager',
   'designer' => 'Designer',
+
  );
+
 $form['job_title'] = array(
+
   '#type' => 'select',
     !#+i+lo! _> +/!lob Ti+lo!
```

Wednesday, May 13, 15

The most important lines are the ones highlighted here.

The plus and minus signs here tell us these lines are changing. "Minus" is a line being removed; "Plus" is a line being added.

On the right, you can see the actual difference that triggered the change.

# When and Why?



Wednesday, May 13, 15

# DON'T HACK CORE!!

#### Why Not?

- Forward compatibility
- Developer Friendliness

#### DON'T DO IT!





In the Drupal Community, we don't hack core! And we really shouldn't hack contrib, either.

There are two main reasons:

1 – Forward compatibility: If new features or changes, or most importantly, security releases, are made to the module, you can't take advantage of them without losing your changes.

2 – Developer friendliness: If your changes introduce a bug down the road, other developers will not look in that module, because they will assume it hasn't been changed. This will cost them time and frustration.

DON'T DO IT!

# Hacking... Nicely.

Sometimes, though... you need to.

Patches are a way to modify core or contrib code in a way that's maintainable, future-friendly, and reproducible.



And it saves kittens.

社

Wednesday, May 13, 15

Sometimes, though... you just need to.

That's where patches come in. They are a way to modify core or contrib code in a way that's maintainable, future friendly, and reproducible.

#### And it saves kittens!

# What's the Difference?

Wednesday, May 13, 15

So, what is the difference between a patch and a hack?

Method.

When I say I "hack" a module, I mean that I am changing the module code directly, putting it straight into my site repo or on my site server, and running it. Changes like this are usually pretty invisible to other developers.

When I say I "patch" a module, it means that the changes that I've made are in a separate text file, which is applied to the module when the site is built. These changes are also easily accessed and reviewed by other developers.

This tiny methodology difference means a great deal in actual practice. A module that's been hacked is very difficult to use in the long term. Changes made to it are often not recorded anywhere (or anywhere anyone would look), and if the module is replaced, by say a new or updated version, then those changes are lost forever.

# When should I patch?

- You've found a module that does most of what you need... but not quite everything.
- You've found a bug in the module.
- You need to integrate custom functionality into the module, but it doesn't have the right API functions.
- You need a change *right now* and the module maintainer isn't responding.



## When should I not?

- The module provides hooks or alter functions that will allow you to do what you need.
- The module only does a little of what you need, and you probably can build a custom module for the same effort.
- The dev version of the module has what you need, or there's already a patch in the issue queue.



Wednesday, May 13, 15

That last point leads us to the next section...

# How?



Wednesday, May 13, 15

## Work Smarter

#### Check the Issue Queue and the Dev Version of the

#### module!

#### Workbench Moderation

#### View Version control Revisions

Posted by becw on March 1, 2011 at 11:07am

Workbench Moderation adds arbitrary moderation states to Drupal core's "unpublished" and "published" node states, and affects the behavior of node revisions when nodes are published. Moderation states are tracked per-revision; rather than moderating nodes, Workbench Moderation moderates revisions.



#### About Workbench

Workbench is a suite of modules which provide easier content management for content administrators. Each of the "Workbench" modules has been tested to work with the main Workbench module, and with the other modules in the Workbench suite. The Workbench suite is modular, allowing site builders to build the workflow that best suits the content administrators on their site. The Workbench suite provides authors, editors, and publishers with a unified interface for managing content relevant to them. It allows people to focus on content, rather than on learning Drupal.

Workbench Moderation may be run as a stand-alone module with no dependencies; however, certain features are available only when the Workbench module is also enabled.

- Workbench
- Workbench Access
- Workbench Moderation Workbench Media
- Workbench Files

#### Sponsors

Development is sponsored by Palantir.net.

#### Maintainers for Workbench Moderation

srjosh - 3 commits last: 1 month ago, first: 1 month ago Dave Reid - 18 commits

last: 11 months ago, first: 3 years ago stevector - 231 commits

last: 1 year ago, first: 3 years ago

robeano - 24 commits last: 1 year ago, first: 2 years ago

agentrickard - 36 commits last: 1 year ago, first: 3 years ago

View all committers View commits

#### Issues for Workbench Moderation To avoid duplicates, please search before submitting a new issue.

Advanced search

All issues

318 open, 616 total Bug report

151 open, 288 total

Wednesday, May 13, 15

Step One: Work Smarter!

It's quite possible that someone has already done what you need to do. Check first, save yourself some work!

If you are not already using the dev version of the module, try that. If that doesn't work...

These links lead to the issue queue, where you can find other people's bug reports, feature requests, support requests, and – best of all – patches.

If you find a patch that looks promising, you can skip to the "Apply a Patch" section towards the end of this presentation.

# Edit and patch



Wednesday, May 13, 15

## Check it out!

Download & Extend Download & Extend Home Drupal Core Distributions Modules Themes	Patch Demo Patch Demo
Patch Demo srjosh's sandbox: Patch Demo	View Edit Version control Maintainers This page gives the essential Git commands for working with this project's source files.
View         Edit         Version control         Maintainers           Posted by srjosh on July 30, 2014 at 10:01pm         Image: Control of the second s	Version to work from * 7.x-1.x * Maintainer? Show     Update Notice: See <u>Git instructions updates</u> for a record of updates to these instructions.
Experimental Project This is a sandbox project, which contains experimental code for developer use only. This is a dama project for use at dama procentations	One-Time Only Setting up repository for the first time
Project Information	git clonebranch 7.x-1.x srjosh@git.drupal.org:sandbox/srjosh/2312963.git patch_demo
Maintenance status: Minimally maintained Development status: Maintenance fixes only	Associating your copy of the repository with your Drupal.org account

[jturton@~/htdocs/presentations/patch\_demo]\$ git clone --branch 7.x-1.x srjosh@git.drupal.org:sandbox/srjosh/2312963.git patch\_demo Cloning into 'patch\_demo'... remote: Counting objects: 5, done. remote: Compressing objects: 100% (5/5), done. remote: Total 5 (delta 0), reused 0 (delta 0) Receiving objects: 100% (5/5), done. Checking connectivity... done



Wednesday, May 13, 15

Step Two: Check out the module!

You need to be working from a git repo to make a patch. Just downloading the module from the links doesn't work.

Find and click the link on the module's page that says "Version Control". Make sure that the "Version to work from" form item says "-x" at the end. This is the dev version of the project.

Then copy and paste the git clone line into your terminal.

## Hack Away!



Now's your chance!

Make changes, experiment, do all the crazy things you can think of.

Don't worry, you're doing it the right way, so the kittens are safe.

Wednesday, May 13, 15

Step Three: The Hack

The magic of source control means that you are not going to wind up with an unusable module – you can always rewind your work and start over if you have to.

Then, enable the hacked module in a site, and test! Repeat this until it works just the way you want.

# Two Things to Remember

- Make sure you're working against the dev version of the module. If you are submitting to the module's issue queue (and we will be) you'll usually be asked by the maintainer to roll your patch against the dev version.
- And please, please make sure you try to follow best Drupal security practices!

Wednesday, May 13, 15

Make sure you're working against the dev version of the module. It's the latest code, and when you submit it to the maintainer, they will expect you to be working on the newest code.

Also – please make sure to follow Drupal's best practices for security! Use Form API, use database abstraction, sanitize your text, and so on. There's lots of good reading on this topic.

# Once you're done, diff.

[jturton@~/htdocs/presentations/patch\_demo/patch\_demo]\$ git diff > patch\_demo.patch

The Command line way:

#### git diff > patch\_demo.patch

Wednesday, May 13, 15

Step Four: make the patch. There's two ways.

The command line way: cd into your module directory

#### git diff > patch file

This will put all modified files into the patch. There are command line options to only include certain files, or to compare different directories, etc. However, I personally favor a different way...

# Once you're done, diff.

000							patch_	demo (Git)			
View Commit	Checkout	Reset Stash	Remove	Fetch F	L T Pull Push	Branch	Merge	Show in Find	er Git Flow	Create Patch Apply Patch	
FILE STATUS		Pending files, s	orted by path							(9.	Search
Working Copy	2	Staged files							(		
BRANCHES								Change	patch_d	filos tes 31-43	Sta
TAGS								Changes	s to those	default' => '',	
REMOTES	Hide								32 32 33 33 34 + 35 + 36 +	<pre>'description' =&gt; 'Surname of the ), 'job_title' =&gt; array(     'type' =&gt; 'varchar',     'length' =&gt; 255,</pre>	person.',
SURMODULES									37 +	'not null' => TRUE,	
SUBTREES .									39 +	'description' => 'Job title of th	e person.',
	ſ	Files with c	hanges						35 42 36 43	'primary key' ⇒ array('pid'), 'indexes' ⇒ array(	
		Unstaged file	is .						parch_o	lunk 1 : Lines 125-131	Sta
	- 11	a patch	demo.install						125 125	<pre>\$rows[] = \$entry_row;</pre>	_
			demo module			_			126 126	<pre>} // Make a table for them.</pre>	
	Ľ	U U Parci	_demo.mousie						128 - 128 + 129 129 138 138 131 131	<pre>Sheader = array(t('Id'), t('Name'), Sheader = array(t('Id'), t('Name'), Soutput .= theme('table', array('he ) else {</pre>	t('Surname'), t('Surname'), ader' ⇒ Shead
									E	lunk 2 : Lines 165-183	Sta
									165 165 166 166 167 167	'#default_value' => !empty(Sdefault_);	_entry) ? \$def
									168 + 169 +	<pre>Soptions = arrayl     'developer' =&gt; 'Developer',</pre>	



Wednesday, May 13, 15

#### Sourcetree.

This is a GUI interface for management of your repositories, and it is awesome. It's also free.

It's made by Atlassian, the same company that makes Jira (ticket management), Confluence (wiki) and BitBucket (repository), so it's reliable, regularly updated, and easy to use.



ES       Hide         igin       33       33       34       1         es       DULES       38       48       37       48         BULES       Files with changes       36       43       14       1         Image: Image of the set of the se					32	32	
igin ES DULES EES Files with changes ↓ Unstaged files ↓ Uns	ES	Hide			33	33	),
ES DULES TES Files with changes Unstaged files Unstaged files patch_demo.module patch_demo.module Tor 125 125 126 126 127 127 127 128 - \$ 129 129 \$ 129 129 \$ 129 129 \$ 129 129 \$ 121 121 121 125 \$ 126 126 \$ 127 127 7/7 128 - \$ 129 129 \$ 120 126 \$ 121 121 \$ 125 125 \$ 126 126 \$ 127 127 7/7 128 - \$ 129 129 \$ 120 126 \$ 125 125 \$ 127 127 \$/7 128 - \$ 126 126 \$ 127 127 \$/7 128 - \$ 129 129 \$ 129 129 \$ 129 129 \$ 129 129 \$ 129 129 \$ 129 129 \$ 120 128	igin					34 +	
DULES       37       +         SES       Files with changes       39       +         Unstaged files       -       100       100         DULES       DULES       DULES       100       100         Files with changes       -       100       100       100         Dults       Dults       -       100       100       100         Dults       Dults       -       100	ES					36 +	
DULES       38 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 48 + 39 + 35 + 42 + 35 + 52 + 52 + 52 + 52 + 52 + 52 + 5						37 +	
Files with changes	DULES					38 +	
Files with changes         Unstaged files            • patch_demo.install         • • • • • • • • • • • • • • •	EES					40 +	).
Files with changes       35 42       'pr:         Unstaged files       image: state in the					34	41	),
Files with changes       35 43       'inc         Unstaged files					35	42	'pri
□ Unstaged files       Hunk 1         □ □ patch_demo.install       125       125         □ □ patch_demo.module       127       127       127         128       -       \$1       \$1       128       +       \$1         129       129       \$2       \$3			Files with changes		36	43	'ind
□       Unstaged files						patch	_demo.m
•••••••••••••••••••••••••••••			Unstaged files				Hunk 1
••••         ••••         ••••			patch_demo.install		125 126	125 126	}
128       -       \$1         129       129       129         130       130       130         131       131       else         165       165       ';         166       166       );         167       167       167         169       +       'c         170       +       'g         170       +       'g		_	patch_demo.module		127	127	11
128       +       \$1         129       129       \$4         130       130       \$3         131       131       \$1         131       131       \$1         165       165       '5         166       166       \$;         167       167       167         169       +       '6         170       +       '5         170       +       '5					128	120	- \$h
130       130       130       131         131       131       131       131       else         165       165       '4         166       166       );       167       167         167       167       167       169       +         169       +       '0       170       +       '170         170       Commit message       Commit message       Commit message       Commit message					129	128	+ \$n \$0
131       1					130	130	}
Hunk 2 165 165 166 166 167 167 168 + \$opti 169 + 'c 170 + 'g					131	131	else
165       165       165       165       165       165       165       165       165       165       165       167       167       167       167       169       +       *       169       +       *       169       +       *       169       +       *       170       +       * </th <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th>Hunk 2</th>							Hunk 2
166       166       166       );         167       167       167         169       +       'c         170       +       'g         Image       Commit message					165	165	'#
167 167 168 + \$opt 169 + 'c 170 + 'g					166	166	);
100     +     \$00       169     +     '0       170     +     '1				L L	167	167	+ Cont
170 + 1						169	+ \$000 + 'd
Commit message						170	+ 'p
			Commit message				
			2 Commit mossage				

Wednesday, May 13, 15 It lists the files you've changed...



Wednesday, May 13, 15

and shows you the changes to those files

	<b>S</b>	Open In Terminal	^жт У
View Commit Checkout	t Reset Stash	Quick Look	anch
LE STATUS	Pending files, s	External Diff	
Working Copy	Staged files	Create Patch	
RANCHES		Apply Patch	
AGS		Add to index Remove	ዕ∺= ∺≪
EMOTES		Unstage From Index	<b>公光</b> —
💂 origin		Add / Remove	て ひ 第 =
TASHES		Stop Tracking	
UBMODULES		Commit	
UBTREES		Reset to Commit	€אמית
		Continue Abort	
		Resolve Conflicts Custom Actions	
	Unstaged file	Log Selected	Σ₩L

Wednesday, May 13, 15

And you can create a patch just like that. The best part of creating a patch in Sourcetree is that it will ask you which files to include, in a nice checkbox menu. Much better than having to specify them on the command line.

# Submit your work

Wednesday, May 13, 15

Step Five: Submit your work.

Create a new issue Advanced :	search E-mail notific Status - Open issues -	ations	Pric ÷ A	ority ( Iny- ‡	Category - Any - 🗘 😂	Version - Any -	\$	Component - Any -	\$
Summary	Status	Priority	Category	Version	Component	Replies	Last updated	Assigned to	Created
Integrate with State Flow Schedule new	Needs review	Normal	Feature request	7.x- 2.x-dev	Code	8 7 new	21 min 22 sec		8 months 3 week
One revision per workflow new	Active	Normal	Support request	7.x-1.3	Code		5 hours 4 min		5 hours 4 min
New rules condition when published for the first time only new	Needs work	Normal	Feature request	7.x- 1.x-dev	Code	2 2 new	8 hours 17 min		11 hours 34 min
Editors can't find unpublished nodes that were created by other editors new	Active	Normal	Bug report	7.x-1.3	User interface	1 1 new	1 day 8 hours		1 month 4 hours
'Workbench moderation: current' views filter does not list all content <mark>new</mark>	Reviewed & tested by the community	Normal	Bug report	7.x- 1.x-dev	Code	26 2 new	1 day 18 hours		2 years 7 months
Unnecessary protection against infinite loops new	Active	Minor	Bug report	7.x- 1.x-dev	Code		1 day 20 hours		1 day 20 hours
Node revision history optimization on large	Needs work	Major	Feature	7.x-	Code	51 3	2 days 8		2 years 6

Wednesday, May 13, 15

This is the Issue Queue of a module. (You should already be familiar with this from searching it for solutions before.)

	Workbench Moderation Issues for Work Create a new issue Advanced	bench Mode	rations	
	Search for	Status		
		- Open issues -		\$
		<u> </u>		
		Search		
	Summary	Status	Priority	Categ
	Integrate with State Flow Schedule new	Needs review	Normal	Featu reque
	One revision per workflow			Suppo
Wednesday, May 13, 15				
Create a new issu	le.			

	Project -
	Workbench Moderation (1077528)
Category *       Priority *       Status *         - Select a valu *       Normal *       Active         Descriptions of the Priority and Status values can be found in the	Version *     Component *     Assigned *       +     - Select a valu +     - Select a value - +     Unassigned +       Issue queue handbook.     -     -     -
Issue tags	
Before adding tags read the issue tag guidelines. Do NOT use tag	s for adding random keywords or duplicating any other fields. Separate terms with a comma, not a space
<ul> <li>Issue summary &amp; relationships</li> </ul>	
Issue summary	
B I AF H2 H3 H4 1 code PHP 🛃 🍓 🗄 🗄 🥐	

Wednesday, May 13, 15

Fill out all the form elements – make the title descriptive, and the summary a good explanation of what the problem is.

#### You are not attaching your file just yet! Because...

# Naming

[project\_name]-[short-description]-[issue-number]-[comment-number].patch

```
[module name] - [short description] - [issue number]
- [comment number] . patch
example:
patch_demo-job_field-2056001-3.patch
```

Wednesday, May 13, 15

You're going to rename your patch file to match the Drupal community's standards. Here's how:

module name - short description of what your patch does - issue node number - comment

#### number .patch

Module name and short description are self explanatory, but let me show you where to find those numbers...

## Issue Number



Issue number is the nid of the issue queue node. You find it in the URL.

Wednesday, May 13, 15

After you've created the issue node in the issue queue, the issue number is the nid of that node. It's in the URL.

## Comment Number



Comment number is the number of the comment on the node, plus 1. **NOT** the cid.

Wednesday, May 13, 15

In order to add a patch to the issue node, you'll need to attach it to a comment. Your comment number will be one more than the highest comment number on the node already. If there are no comments, then it will be 1.

#### It is NOT the cid.

# Naming

[project\_name]-[short-description]-[issue-number]-[comment-number].patch

```
[module name] - [short description] - [issue number]
- [comment number] . patch
example:
patch_demo-job_field-2056001-3.patch
```

Wednesday, May 13, 15

Now that you have the numbers, you can rename your patch. In our example here, the patch is to the "patch\_demo" module (a sandbox module I made), the description is "job\_field", because it's a patch to add a field for "jobs" to a user, the issue nid is 2056001, and there were already 2 comments in the issue so this will be attached to #3.

Rename it, then upload and make a comment like "Patch attached".

# Congratulations!





Wednesday, May 13, 15

**Congratulations!** 

You are now a contributing member of the Drupal Community!

#### Think of all the kittens you've saved!

# Use your work in production

Wednesday, May 13, 15

Step Six: Bring all the pieces together.

## Drush Make

core = 7.x	
api = 2	
: Drupal C	bre .
projects[d	rupal][type] = core
projects[d	rupal][version] = 7.30
; If all p	ojects or libraries share common attributes, the 'defaults'
; array ca	be used to specify these globally, rather than per-project.
defaults[p	<pre>rojects][subdir] = "contrib"</pre>
: Contrib	
projects[fe	eatures][version] = 2.0
projects[p	<pre>itch_demo][type] = module</pre>
projects[p	<pre>itch_demo][subdir] = contrib</pre>
projects[p	<pre>itch_demo][download][url] = http://git.drupal.org/sandbox/srjosh/2312963.git</pre>
projects[p	<pre>itch_demo][download][type] = git</pre>
projects[page = page =	<pre>stch_demo][download][branch] = 7.x-1.x</pre>

Wednesday, May 13, 15

Drush make allows you to specify modules, libraries, themes... it makes site building work the way it's supposed to. It's great, and if you aren't using it, you should be.

One reason to use it is, in combination with some other tools, it allows you to avoid

#### committing contributed modules and core code to your repository.



Wednesday, May 13, 15

It also allows you to specify patches to apply to modules. The patches must be somewhere accessible via the web, which is part of why we put it into the Drupal issue queue.

Running drush make on this make file will download drupal, the features module, and our patch\_demo module. It will assemble the site, then apply the patch to the patch\_demo module, making the changes we specified much earlier in the process.

That's all there is to it.

# Community Response

Wednesday, May 13, 15

Discussion of possible outcomes for your patch (Acceptance / Rejection / Discussion).

# Why bother?



Wednesday, May 13, 15

Since the patch only needs to be accessible to drush, it could be anywhere – like, on a file server or your company's web site download directory. Why contribute?

Because each of you, no matter how new, or inexperienced, or totally ninja, can help make Drupal better. Communities work best when everyone contributes. Your patch may not be accepted, but it may, or it may spark a discussion that leads to a better way of doing what you need.

Also – even it it isn't accepted, it's in the issue queue file system, so it never goes away – meaning you can continue to use it. I've submitted many patches I knew would never get accepted, because I needed something to work in a particular unique way. That's OK.

Finally, on a more personal note, contributing in the issue queues helps get you known in the Drupal community. This is great for your career.

# Links

- Drupal Patch Contributor Guide: drupal.org/node/1054616
- joshuabrauer.com/2008/07/dont-hack-core-and-dont-hackcontrib-unless-you-have
- *Git Diff*: git-scm.com/docs/git-diff
- *SourceTree*: sourcetreeapp.com
- Phase2 Blog: phase2technology.com/blog/how-and-whyto-patch-a-drupal-module



• Session Feedback: events.drupal.org/node/775

### Joshua Turton Senior Developer



Email: jturton@phase2technology.com
Twitter: @sjinteractive

Wednesday, May 13, 15

Thank you all so much for coming! Again, I'm Joshua Turton with Phase2, and you can reach me at this email or twitter handle, or at our booth in the exhibitor's hall.

Please, if you liked this talk, or even if you didn't, leave feedback on the session page! It's

#### also where you will be able to download the slides.

If you'd like a demo of this process, find me at our booth and we can walk through it together!

# Drupal Camp Costa Rica





July 29<sup>th</sup> through 31<sup>st</sup> Universidad de Costa Rica



#### drupalcamp.cr/en

Wednesday, May 13, 15

One last thing, a plug for Drupal Camp Costa Rica, happening in San Jose, Costa Rica at the end of July. If you are interested in speaking, sessions submissions will be open soon! Sponsorship opportunities will also be available. Talk to me for more information.



PHASE2TECHNOLOGY.COM

Wednesday, May 13, 15 Questions?