

How and Why to Patch a Module

November 8, 2014



Sunday, November 9, 14

Good Afternoon!
[Introduction and overview.]

What's a Patch?



A Small Change File

Patches are small files that contain a list of all the differences between the publicly-available code, and the code you want to run.

You “apply” them to the existing code, and your changes are made in the code.



Sunday, November 9, 14

They are a small file that contains a list of all the differences between the existing code, and the code you want to run.

This allows you to both use the code you want, and also keep up with changes in the contrib or core code you're basing it. Every time they update, you can re-run your patch on the new version, to modify the module and make it do what you want it to.

Why and When?



Hacking core/contrib... Nicely

In the Drupal community, we

DON'T HACK CORE!

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

And it saves kittens.



Sunday, November 9, 14

There are two main reasons we don't hack core or contrib.

- 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!

When would I patch a module?

- 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 would I *not* patch a module?

- 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.



How?



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Workbench Moderation adds arbitrary moderation states to Drupal core's "unpublished" and "published" node states, and affects the behavior of node revisions when nodes are published.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
- [stevevector](#) – 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.

[Search](#)

[Advanced search](#)

[All issues](#)
318 open, 616 total

[Bug report](#)
151 open, 288 total

[Subscribe via e-mail](#)



Sunday, November 9, 14

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



Check it out!

Download & Extend

[Download & Extend Home](#) [Drupal Core](#) [Distributions](#) [Modules](#) [Themes](#)

Patch Demo

srjosh's sandbox: Patch Demo

[View](#) [Edit](#) [Version control](#) [Maintainers](#)

Posted by [srjosh](#) on July 30, 2014 at 10:01pm

Experimental Project
This is a [sandbox project](#), which contains experimental code for developer use only.

This is a demo project for use at drupal camp presentations.

Project Information

Maintenance status: [Minimally maintained](#)
Development status: [Maintenance fixes only](#)

Patch Demo

[View](#) [Edit](#) [Version control](#) [Maintainers](#)

This page gives the essential Git commands for working with this project's source files.

Version to work from * ☒ Maintainer? [Show](#)

- Update Notice: See [Git instructions updates](#) for a record of updates to these instructions.

One-Time Only

Setting up repository for the first time

```
git clone --branch 7.x-1.x srjosh@git.drupal.org:sandbox/srjosh/2312963.git patch_demo
cd patch_demo
```

Not working for you? See [Troubleshooting Git clone](#).

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
```



Sunday, November 9, 14

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.



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!



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
```



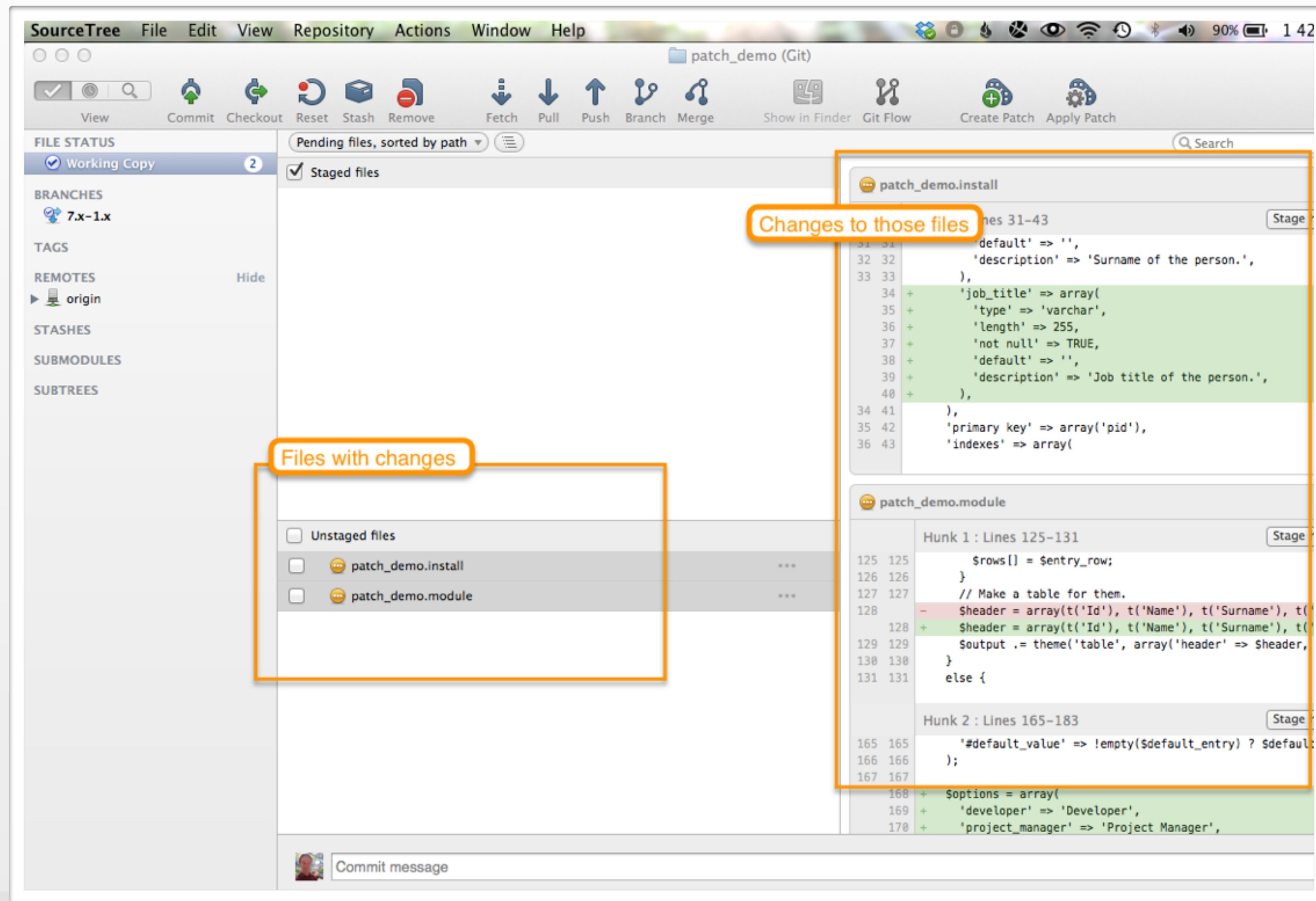
Sunday, November 9, 14

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.



Sunday, November 9, 14

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

ODULES

EES

Files with changes

☐ Unstaged files

☐ patch_demo.install

☐ patch_demo.module

patch_demo.m

Hunk 1

125 125
126 126
127 127
128
128
129 129
130 130
131 131

- \$h

+ \$h

\$o

}

else

Hunk 2

165 165
166 166
167 167
168
168
169
170

'#

);

\$opt

'd

'p

Commit message

Sunday, November 9, 14

It lists the files you've changed...

Merge Show in Finder Git Flow Create Patch Apply Patch Search

patch_demo.install

Changes to those files

Lines 31-43

```
31 31         'default' => '',
32 32         'description' => 'Surname of the person.',
33 33     ),
34 +     'job_title' => array(
35 +         'type' => 'varchar',
36 +         'length' => 255,
37 +         'not null' => TRUE,
38 +         'default' => '',
39 +         'description' => 'Job title of the person.',
40 +     ),
34 41     ),
35 42     'primary key' => array('pid'),
36 43     'indexes' => array(
```

patch_demo.module

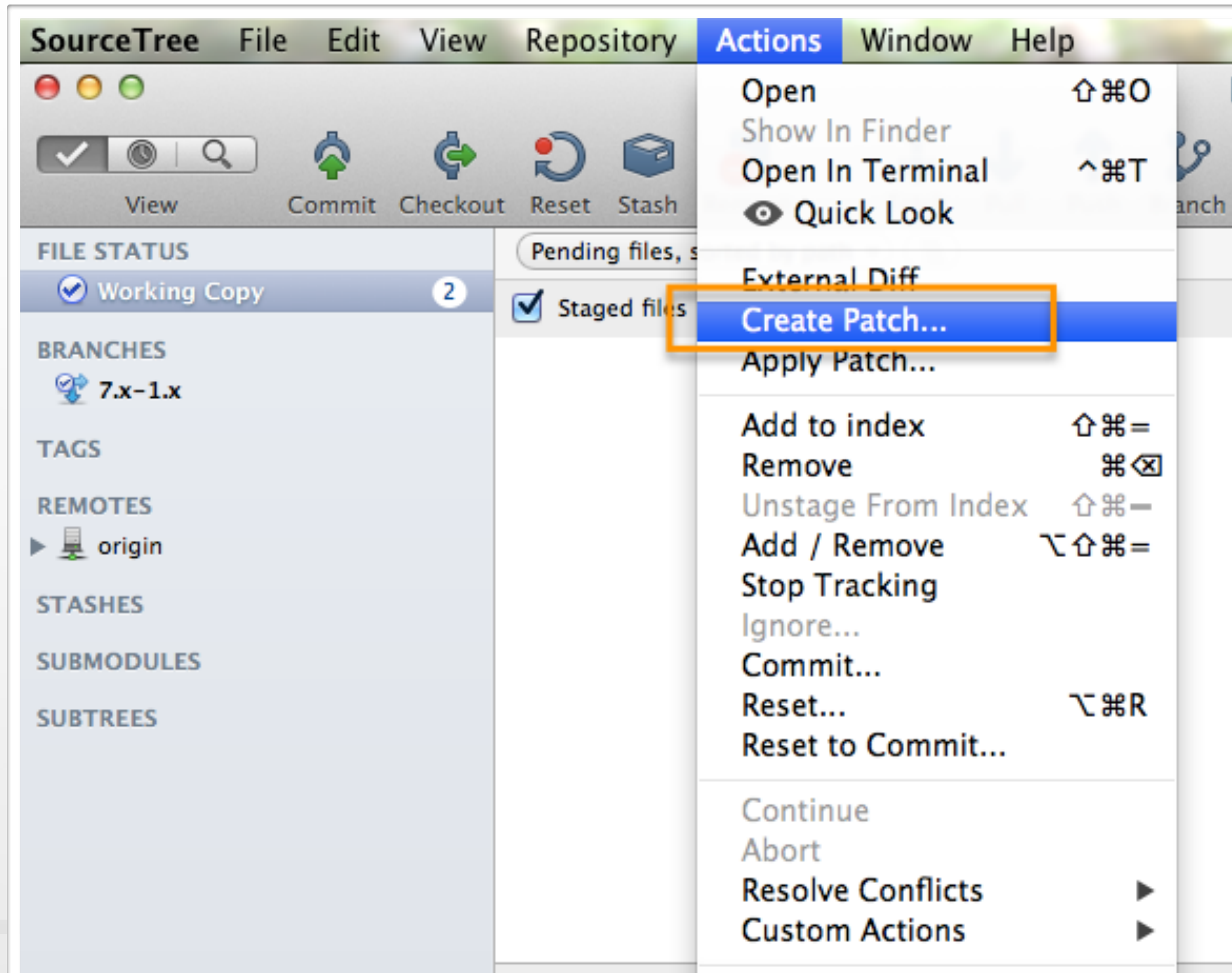
Hunk 1 : Lines 125-131

```
125 125     $rows[] = $entry_row;
126 126 }
127 127 // Make a table for them.
128 - $header = array(t('Id'), t('Name'), t('Surname'), t('
128 + $header = array(t('Id'), t('Name'), t('Surname'), t('
129 129 $output .= theme('table', array('header' => $header,
```



Sunday, November 9, 14

and shows you the changes to those files



Sunday, November 9, 14

And you can create a patch just like that. The best part of creating a patch in Sourcetree is that if you have changed more than one file, 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



Issues for Workbench Moderation

[Create a new issue](#)[Advanced search](#)[E-mail notifications](#)

Search for

Status

- Open issues -

Priority

- Any -

Category

- Any -

Version

- Any -

Component

- Any -

[Search](#)

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 weeks
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 new	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 request	7.x-1.x-dev	Code	51 3 new	2 days 8 hours		2 years 6 months



Workbench Moderation

Issues for Workbench Moderation

Create a new issue

Advanced search

E-mail notifications

Search for

Status

– Open issues –

Search

Summary	Status	Priority	Categ
Integrate with State Flow Schedule new	Needs review	Normal	Featur reque

Sunday, November 9, 14

Create a new issue.

▼ Issue metadata

Title *

Project *

Workbench Moderation (1077528)

Category *

– Select a value –

Priority *

Normal

Status *

Active

Version *

– Select a value –

Component *

– Select a value –

Assigned *

Unassigned

Descriptions of the [Priority](#) and [Status](#) values can be found in the [Issue queue handbook](#).

Issue tags

Before adding tags [read the issue tag guidelines](#). Do **NOT** use tags for adding random keywords or duplicating any other fields. Separate terms with a comma, not a space.

▼ Issue summary & relationships

Issue summary

B

I

ABC

H2

H3

H4

code

PHP

[More information about text formats](#)

Sunday, November 9, 14

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

Don't attach your file yet!

Naming

```
[project_name]-[short-description]-[issue-number]-[comment-number].patch
```

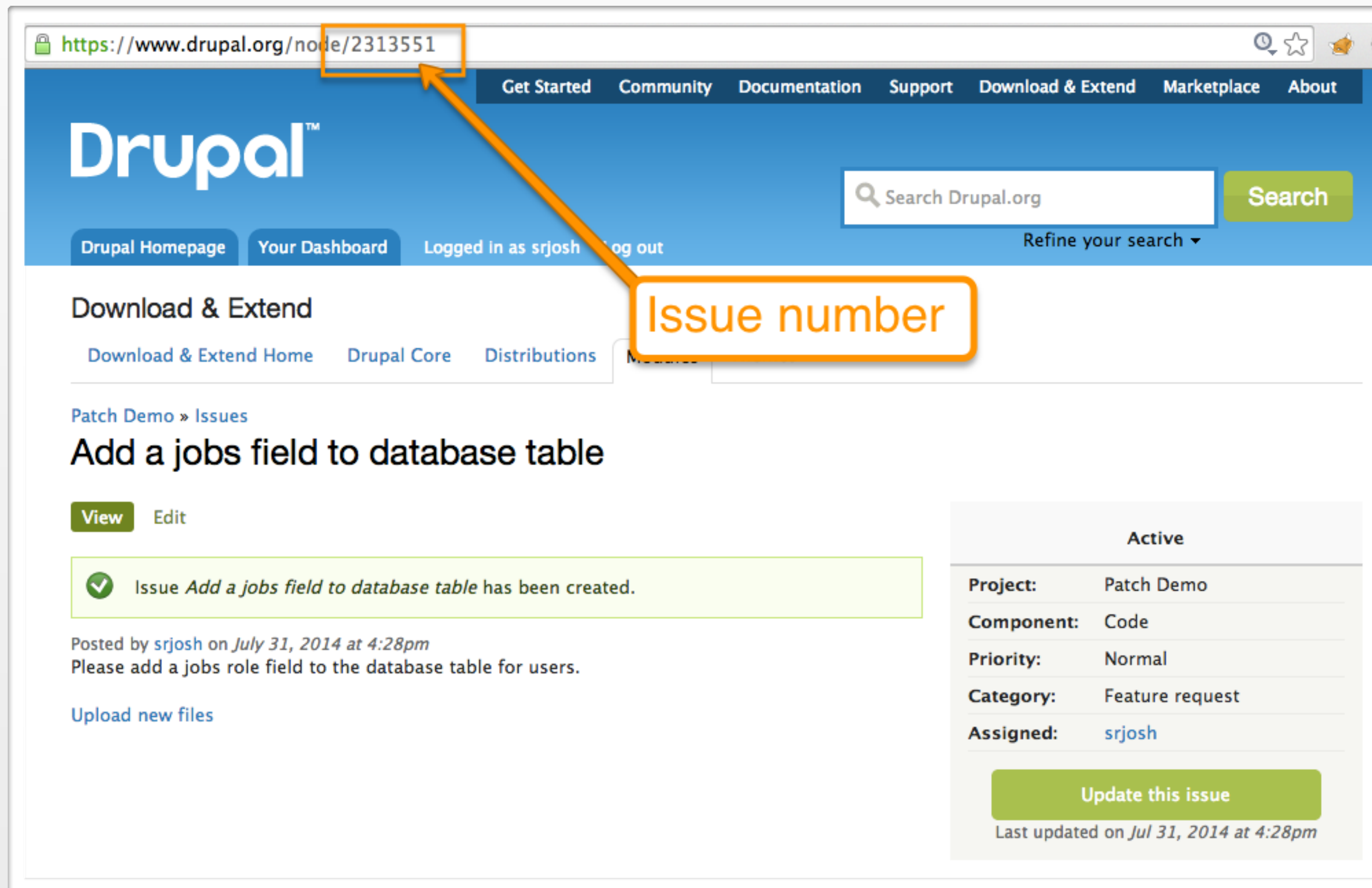
[module name] - [short description] - [issue number] -
[comment number] . patch

example:

patch_demo-job_field-2056001-3.patch



Issue Number



The screenshot shows a Drupal issue page. The URL in the browser's address bar is `https://www.drupal.org/node/2313551`, with the number `2313551` highlighted by an orange box and an arrow pointing to it. Another orange box labeled "Issue number" points to the same number in the URL. The page title is "Add a jobs field to database table". A green message box states: "Issue Add a jobs field to database table has been created." The issue was posted by `srjosh` on July 31, 2014 at 4:28pm. The issue details table shows: Project: Patch Demo, Component: Code, Priority: Normal, Category: Feature request, Assigned: `srjosh`. There is an "Update this issue" button and a note "Last updated on Jul 31, 2014 at 4:28pm".

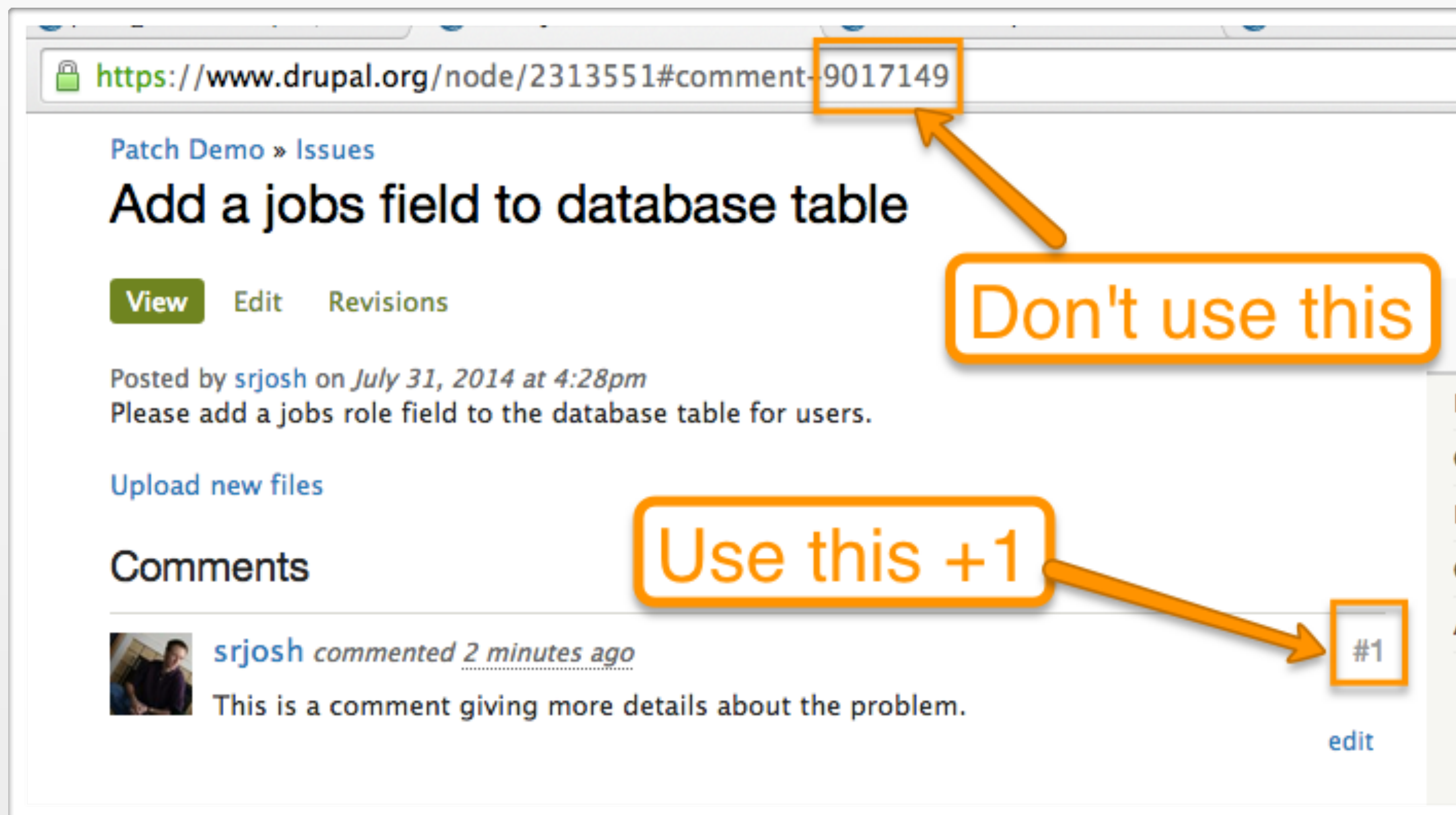
Issue number is the
nid of the issue queue
node.

You find it in the URL.



Comment Number

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



Sunday, November 9, 14

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



Use your work in production



Drush Make

```
core = 7.x
api = 2

; Drupal Core
projects[drupal][type] = core
projects[drupal][version] = 7.30

; If all projects or libraries share common attributes, the `defaults`
; array can be used to specify these globally, rather than per-project.
defaults[projects][subdir] = "contrib"

; Contrib
projects[features][version] = 2.0

projects[patch_demo][type] = module
projects[patch_demo][subdir] = contrib
projects[patch_demo][download][url] = http://git.drupal.org/sandbox/srjosh/2312963.git
projects[patch_demo][download][type] = git
projects[patch_demo][download][branch] = 7.x-1.x

projects[patch_demo][patch][2313551] = http://drupal.org/files/patch_demo-job_field-2313551-2.patch
```




```
api = 2

; Drupal Core
projects[drupal][type] = core
projects[drupal][version] = 7.30

; If all projects or libraries share common attributes, the `defaults`
; array can be used to specify these globally, rather than per-project.
defaults[projects][subdir] = "contrib"

; Contrib
projects[features][version] = 2.0

projects[patch_demo][type] = module
projects[patch_demo][subdir] = contrib
projects[patch_demo][download][url] = http://git.drupal.org/sandbox/srjosh/2312963.git
projects[patch_demo][download][type] = git
projects[patch_demo][download][branch] = 7.x-1.x

|projects[patch_demo][patch][2313551] = http://drupal.org/files/patch_demo-job_field-2313551-2.patch
```



Sunday, November 9, 14

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

Note that, by default, the file link from the issue queue will be https; we've run into issues with that and I recommend always removing the 's' from the URL.

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



Sunday, November 9, 14

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

Why bother?



Sunday, November 9, 14

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 if 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-hack-contrib-unless-you-have
- Git Diff: git-scm.com/docs/git-diff
- SourceTree: sourcetreeapp.com
- Phase2 Blog: www.phase2technology.com/blog/how-and-why-to-patch-a-drupal-module



Joshua Turton

Senior Developer

Email: jturton@phase2technology.com

Twitter: [@sjinteractive](https://twitter.com/sjinteractive)





PHASE2TECHNOLOGY.COM
