

How to figure out which commit causes a bug

Something worked. And now it doesn't. This is called a **regression**.

This happens. Most commits fix/improve things, but every once in a while, something breaks. Sometimes, by fixing a bug, we introduce another. Tiki has so many features that it's difficult to test everything. Lots of eyeballs and good bug reporting is essential

You'd like to report the bug and inform the right person. But how to figure out who introduced the problem and when?

If you are using Git

The following actions must be done on a working copy (usually local) of your Tiki after all commits and conflicts are solved (check with `git status` if everything is up-to-date)

You can use `git bisect` to find out what commit is the cause of the regression.

The `git bisect` command requires just two information to start:

1. A revision where things were definitely good.
2. A revision where the bug is present.

Now, `git bisect` will help to track the bug somewhere between the "bad" and "good" commits. It will splits the range of commits between "good" and "bad" in half - and checks out a commit in the middle.

For each iteration you will have to test and confirm if the bug is still there or gone.

In practice (sample)

Go in your terminal, shell, etc. and enter the following where `10df1cb84df250eaae72cef31aca79a32dffef87` is a commit where the bug is present; **bad** and `165cc018e5dd74485b7e6aa13b53361db1fb3596` is a commit where the bug is not present; **good** and `.`. Note you can use the number or the commit SHA.

start Git Bisect

```
git bisect start
```

declare the commit that has the bug

```
git bisect bad 10df1cb84df250eaae72cef31aca79a32dffef87 //If it is the last version you can use HEAD
```

declare the commit that doesn't have the bug

```
git bisect good 165cc018e5dd74485b7e6aa13b53361db1fb3596
```

The output from the last command should look like this:

```
Bisecting: 3 revisions left to test after this (roughly 2 steps)
[e0375ad20652b1c5974a634809b269d68d6902a5] [FIX] bs4: Use btn-secondary for tracker item
preview buttons, and btn-link for cancel/close like wiki page edit. (missed from 390f4082,
thanks @Jyhem
```

Now you need to go back to you browser and check if, using this version, the bug is there or not.
If it is there continue tracking using

continue the search

```
git bisect bad
```

If the bug is not there continue tracing using

continue the search

```
git bisect good
```

It will repeat the process (splitting commits in half and on) until you are able to successfully singled out the bad commit!

Once it is done don't forget to reset your working copy

stop Git Bisect and reset

```
git bisect reset
```

The output should be something like that:

```
Previous HEAD position was e0375ad206 [FIX] bs4: Use btn-secondary for tracker item preview
buttons, and btn-link for cancel/close like wiki page edit. (missed from 390f4082, thanks
@Jyhem)
Switched to branch 'bernardsfez_23x'
Your branch is up to date with 'origin/bernardsfez_23x'.
```

If you are using SVN

Please note: SVN updates (for all versions of Tiki) will stop in January 2023 when [Tiki18](#) reaches end of life, so you should get your updates from Git

You can use [SVN-bisect](#) or the manual procedure below. (Could <http://code.google.com/p/phpsvnclient/> be used as well?)

1. You should subscribe to [SVN mailing list](#) to be informed of all fixes and to search in your mail client
 - This is a high-volume list so you may want to put a filter to store messages in a directory
2. [Get Tiki via SVN](#) (choose your branch)

3. Check your version number.

svn info

```
$ svn info
Path: .
url: http://svn.code.sf.net/p/tikiwiki/code/branches/7.x
Repository Root: http://svn.code.sf.net/p/tikiwiki/code/
Repository UUID: b456876b-0849-0410-b77d-98878d47e9d5
Revision: 34719
Node Kind: directory
Schedule: normal
Last Changed Author: jonnybradley
Last Changed Rev: 34716
Last Changed Date: 2011-05-31 10:42:10 -0700 (Tue, 31 May 2011)
```

4. Update SVN to a specific revision which you know works.

To update your source to a *specific revision*, use:

Updating to revision 34715

```
[demotw@alpha 7x]$ svn update -r 34715
U   lib/categories/categlib.php
U   templates/categobjects_title.tpl
U   templates/categobjects.tpl

Fetching external item into 'lib/adodb'
Updated external to revision 34668.

Fetching external item into 'lib/smarty'
Updated external to revision 34668.

Fetching external item into 'lib/jscalendar'
Updated external to revision 34668.

Fetching external item into 'lib/htmlpurifier'
Updated external to revision 34668.

Fetching external item into 'lib/pclzip'
Updated external to revision 34668.

Fetching external item into 'lib/jquery'
Updated external to revision 34668.

Fetching external item into 'lib/jquery/jquery.s5'
Updated external to revision 41.

Fetching external item into 'lib/jquery/jquery.sheet'
Updated external to revision 34668.

Fetching external item into 'lib/jquery/jquery.sheet/jquery-ui/ui'
Updated external to revision 4075.

Fetching external item into 'lib/jquery/jquery.sheet/jquery-ui/theme'
Updated external to revision 4075.
```

```
Updated to revision 405.  
Updated to revision 34668.  
  
Fetching external item into 'lib/phpcas'  
Updated external to revision 23921.  
  
Fetching external item into 'lib/ezcomponents'  
Updated external to revision 34668.  
  
Fetching external item into 'lib/ckeditor'  
Updated external to revision 34668.  
  
Fetching external item into 'lib/codemirror'  
Updated external to revision 34668.  
  
Fetching external item into 'lib/svg-edit'  
Updated external to revision 2034.  
  
Fetching external item into 'lib/mobileesp'  
Updated external to revision 34668.  
  
Fetching external item into 'lib/html5shim'  
Updated external to revision 34668.  
  
Fetching external item into 'lib/pear/Text'  
Updated external to revision 34720.  
  
Fetching external item into 'lib/core/Zend'  
Updated external to revision 24101.  
  
Updated to revision 34715.  
[demotw@alpha 7x]$
```

5. Clear all caches by visiting `tiki-admin_system.php?do=all`
6. Test to confirm the bug is not present
7. Update SVN to a specific revision which you know doesn't work.
8. Test to confirm the bug is present
9. Repeat by changing revision number back & forth until you can pinpoint the revision which introduces the issue.

Since you may have quite a few trial and errors, you may want to ignore externals, but an issue can be caused by an external...

```
svn update --ignore-externals -r 34750
```

* externals causing regressions should be gradually minimized as we tie our externals to specific stable versions

10. Reply to the message on the [SVN mailing list](#) to specify what issue this commit is introducing.

Note: Database structure changes between version, so you may see bugs/error messages because of this. However, most of the time, it won't prevent you to use this method to pinpoint the problematic commit.

SVN-Bisect

How to install SVN-bisect on Ubuntu

```
$ sudo apt-get install svn-bisect
```

Tell svn-bisect good bad, until it gets the right commit number

```
# sudo svn-bisect start 43700 43800
# svn-bisect bad
# svn-bisect good
# svn-bisect bad
# svn-bisect bad
# svn-bisect good
```

Ref:
<http://manpages.ubuntu.com/manpages/lucid/en/man1/svn-bisect.1.html>

Related

- <https://mozilla.github.io/mozregression/>
- <http://www.selenic.com/mercurial/hg.1.html#bisect>
- <https://launchpad.net/bzr-bisect>
- <http://www.monotone.ca/docs/Bisecting.html>
- <http://www.fossil-scm.org/xfer/help/bisect>
- [Subversion](#)
- <https://quality.mozilla.org/docs/bugzilla/guide-to-triaging-bugs-for-firefox/finding-a-regression-window/>

alias

- [Bisect](#)
- [SVN-Bisect](#)
- [SVN Bisect](#)
- [Git Bisect](#)
- [How to figure out what revision number causes a bug](#)