Commit Code

Note: Git is now the preferred method for contributing to Tiki, please see the Git Workflow for instructions (and update them where necessary).

You can now simply make a merge request at https://gitlab.com/tikiwiki/tiki/merge_requests thanks to the Git and SVN combined workflow

How to Submit Tiki Code Changes

Here are step-by-step instructions on how to submit Tiki code changes under two different situations: when using a hosted server and when using your own PC as a server for development purposes. How to get commit access and Where to commit are recommended reading before starting these instructions.

Although the process is a little different for each situation, the basic idea is the same. First, you **check out** the developer version you wish to make changes to. Checking out means downloading the entire developer version from the **repository** (SourceForge.net) to your server using a **SVN** program. An SVN program is used because it will keep track of the changes you make and allow you to upload only the changed files to the repository. The entire Tiki program is downloaded (not just the files you want to change) so that you can install and test the Tiki version with your changes. After making and testing your changes, you **commit** them to the repository using the SVN program, which means your changed files are uploaded into the repository, creating a new Tiki development version that includes your changes.

Below, the processes for both situations (hosted server and PC as the server) are described.

Prerequisites

In Either Situation

The following are must-haves in order to commit whether you are using a hosted server or your own PC as the server:

- **SourceForge Account:** You will need to have an account at SourceForge.net - click here to register for one.

- **Project Administrator Permission:** A Tiki Project Administrator will need to grant you permission to write to the code repository by adding your SourceForge username to the Tiki project as a developer. Project Administrators are listed at http://sourceforge.net/projects/tikiwiki/develop.

- **Decide Version:** There are different developer versions that changes can be made to. Before you begin, decide or get advice on which version you should download and make changes to. See Where to commit for more information. Which version you intend to change affects the path you will need to specify when checking out and downloading those files. The paths for the different versions can be found at SourceForge.net at https://svn.code.sf.net/p/tikiwiki/code/.

With a Hosted Server

The following are must-haves in order to commit on a hosted server in addition to the general prerequisites noted above:

- **SSH/Shell Access:** You will need to access your server using shell access. This may be available from your web host (or you may be able to request it). If so, it will show up on the cPanel (if you have one) in the Security section. Otherwise, there are free third-party programs that do the same thing (like PuTTy - and here's a list of such programs).
- **SVN:** Your server will need to have SVN installed. If it's not installed, try typing `apt-get install subversion` while connected through shell access to install it.

With Your PC as Server

In addition to the must-haves noted above for either situation, here are additional must-haves in order to commit when using your PC as a server:

- **Third-Party SVN Program:** You will need to install a third-party SVN program. A list of such programs can be found on this Wikipedia page, or see the Setup a Development Environment page on this site which links to pages documenting how to create a development environment.

- **Third-Party source code editor:** You will need to download and install a third-party code editor to make the changes you wish to make. Do not use a plain text editor (like Word). A list of code editors can be found on this Wikipedia page, or see the Setup a Development Environment page on this site for some tips.

Of course other software is required to set up your PC as a server in the first place (see Using Your PC as the Server on the next page or see Setup a Development Environment for some tips.

Using a Hosted Server

Overview of Steps

Here's an overview of the steps for committing using a hosted server:

- Get onto your server through **SSH/Shell Access**
- Run the **svn** command to download the development version you want to change onto your server
- Make the code changes you want to make and save them
- Run the **svn** command to check your changes
- Test your the Tiki install that you just changed to see if it functions
- Run the **svn** command to commit your changes
- End the **SSH/Shell Access** connection

These steps are described in detail below.

Step 1: Get on Your Server Using SSH/Shell Access

1. Access the server you want to download to through SSH/Shell Access (through cPanel or third-party software). You will need to enter the following at a minimum:
    1. The server's host name or IP Address
    2. Your username on that server
    3. The related password

    Connect to remote server via SSH

    ```
    ssh username@server.com enter password
    ```

    You can accept message about RSA key fingerprint.

2. Once on your server, move to the web directory

    Move to the web directory, typically "www"

    ```
    cd www
    ```

3. Create an empty directory on your server to download to like you would for a normal install of Tiki

    Create a directory, for example "tiki"

    ```
    mkdir tiki
    ```

4. Move to the directory you want to download to. For example, if you want to download to **tiki**, then type the following and hit enter:

    Move to download directory

    ```
    cd tiki
    ```

Step 2: Checkout Files via SVN

Type in `svn checkout (path) .` and hit enter to check out a development version. The (path) in the command is the path on SourceForge.net to the version you wish to make changes to.

Below, two alternatives are shown: checking out a branch and checking out the trunk. In both cases, assume we found a bug in the last_mod plugin and therefore want to change the `lib/wiki-plugins/wikiplugin_lastmod.php` file.

1. Example: Checking out Branch 12.x (see Get code for available Versions)

    Checking out branch 12.x

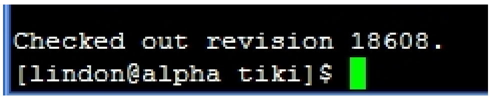
    ```
    svn checkout https://svn.code.sf.net/p/tikiwiki/code/branches/12.x .
    ```

2. Example: Checking out Trunk

Checking out trunk

```
svn checkout https://svn.code.sf.net/p/tikiwiki/code/trunk .
```

After hitting enter, you should see the file names scrolling up the screen as the files are downloaded. At the end, you should see a message telling which revision has been checked out, as in this screenshot:



```
Checked out revision 18608.
[lindon@alpha tiki]$
```

Step 3: Make Your Changes

You can make changes either using your normal edit interface or through the shell access.

Using Normal Edit Interface

The easiest way to make changes is usually through whatever interface you normally use, for example by going through **cPanel > File Manager** and opening the file with your server's code editor, however using a editor like PhpStorm will save immeasurable amounts of time if moderate changes are made. You may also want to familiarize yourself with Tiki's DevTips and Secure Coding Practice Guidelines. After you've made and saved your changes, go to **Step 4**.

Expand the instructions below if you're making changes through the shell access instead.

Use Shell Access to Make Changes

[+]

Step 4: Check Your Changes

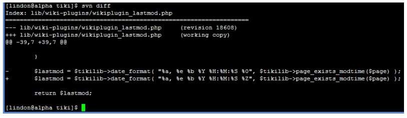
1. Go back to the SSH/shell access screen and make sure you are in the top level tiki directory, **www/tiki** in our example

2. In the SSH/shell access screen type `svn diff` to view your changes:

   Use svn diff to view changes

   ```
   svn diff
   ```

   All changes you have made to all files will be shown as in the following screenshot:

   ```
   [lindon@alpha tiki]$ svn diff
   Index: lib/wiki-plugins/wikiplugin_lastmod.php
   ===================================================================
   --- lib/wiki-plugins/wikiplugin_lastmod.php     (revision 18608)
   +++ lib/wiki-plugins/wikiplugin_lastmod.php     (working copy)
   @@ -39,7 +39,7 @@

           }

   -       $lastmod = $tikilib->date_format( "%a, %e %b %Y %H:%M:%S %O", $tikilib->page_exists_modtime($page) );
   +       $lastmod = $tikilib->date_format( "%a, %e %b %Y %H:%M:%S %Z", $tikilib->page_exists_modtime($page) );

           return $lastmod;

   [lindon@alpha tiki]$
   ```

   As you can see, only one line was changed here (actually one character, an O was changed to a Z). Since this was the intention, we will move on to test the change.

Step 5: Test Your Tiki Install

1. If you haven't already, install the developer version you have downloaded and changed by following the normal Installation procedures through an internet browser.

2. Once installed, test Tiki to ensure that your changes work properly.

Step 6: Commit Your Changes

1. Type the `svn commit` command which consists of 3 parts:

    1. The svn command: `svn commit -m`

    2. A note describing the change, for example: `[FIX] (short description of fix)`
       Your commit message may start with one or more to distinguish changes. Descriptions with tags will go into the changelog and tags will help to classify.

The following tags can be used to distinguish changes:

- NEW for additions of new features
- ENH is an enhancement; not really new but makes things work better (e.g. look and feel, performance...)
- DOC for improvements in the tips or contextual documentation provided to the user/admin
- FIX for bug fixes of any sort
- SEC for security fix operations. If you feel you have reason to put this tag, please first contact security at tiki.org.
- UPD for updates of third party/vendor libraries
- UX for user experience improvements; makes Tiki easier to use and understand (more details in https://dev.tiki.org/UX)
- DB for changes in the database
- MOD is a change which may be disruptive. For example, changing the default value of an option.
- REM for feature removals
- REF for refactoring; changes the structure of the code (to make it cleaner or clearer), without changing its actual behaviour.
- KIL for removals of unused or obsolete files. This tag was used in the sense of REM prior to Tiki 6.
- REL for the release process
- MRG for branch merges, generally performed by the merge scripts
- TRA for translation

When possible, it's also nice to indicate what feature is concerned by the change.
The tags info is also online: https://dev.tiki.org/Commit+Tags

++ If there is a report of the issue solved, add a line to indicate it. For example, if the change addresses issue #426: "Ref: http://dev.tiki.org/wish426" ++ ++ In the case of a "**backport**," (committing a change already made in trunk to an earlier version), a backport reference should be added in front. For example, `[bp/r61102] [FIX]...` in a commit message to the branch *branches/16x*, means this fix is a backport of trunk revision 61102 to the next version of Tiki 16x. Depending on the period, backports are subject to an extra level of review so please read Quality Team and Where to commit first or ask on the mailing list or IRC if unsure. ++ ++ In the case of a "**rolling back**," (reverting a commit), a "rb" reference should be added in front. For example, `[rb/r61102] ...` in a commit message to the branch *branches/16x*, means the change committed in revision 61102 are rolled back (removed, reverted) to the previous state, before r61102. ++ ++ In the case of a typo fix only and/or while doing code reformatting that won’t need to be in the changelog.txt file it is ok to commit without a tag. ++

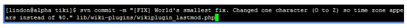
    1. The file path: **filepath**
        - You can leave the file path off, in which case all changes are shown in *svn diff* will be committed (even if more than one file has been changed)

1. Following our example, this would be:

   Commit command

```
svn commit -m "[FIX] Fixed display of time zone, which was showing as %O" lib/wiki-plugins/wikiplugin_lastmod.php
```

Here's a screenshot (with a slightly different description):



```
[lindon@alpha tiki]$ svn commit -m "[FIX] World's smallest fix. Changed one character (O to Z) so time zone appe
ars instead of %O." lib/wiki-plugins/wikiplugin_lastmod.php
```

You may be prompted for a password based on your server username instead of your SourceForge user name. Just hit enter to move to a prompt for entering your SourceForge user name.{REMARKSBOX}

2. You will get a message indicating that your commit was successful

Step 7: Disconnect Shell Connection

Shell command

```
exit
```

Step 8: Be Proud of Yourself

Tiki is the collective work of hundreds of people. It works because of **volunteers**, like you, take the time to make it better.

Notice of your commit will appear in the Tiki IRC chat channel. Go here to see it in the logs or, better yet, log on to the actual chat channel just before you commit and see the notice pop up live!

See next page for instructions when using your own PC as a development server.

...page...

Using Your PC as the Server

These instructions apply when you will be downloading the development version to your own PC which is set up as a server for purposes of developing. Windows and MAC PCs can also be set up as servers when the necessary server programs (Apache, MySQL, PHP, etc.) are installed either individually or packaged together (see this Wikipedia page for a list of such packages).

Overview of Steps

Here's an overview of the steps for committing when using your PC as a server:

- Use a third-party **SVN program** to check out the development version you want to change onto your PC
- Use a third-party **code editor** to make the code changes you want to make and save them
- Use a third-party **SVN program** to check your changes
- Test your the Tiki install that you just changed to see if it functions
- Use a third-party **SVN program** to commit your changes

These steps are described in detail below.

Step 1: Checkout Files Via SVN Program

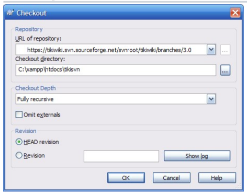
Checking out means you are downloading the files to your PC under a third-party SVN program that keeps track of all changes that are made.

Below, assume that we found a bug in the last_mod plugin and therefore want to change the `lib/wiki-plugins/wikiplugin_lastmod.php` file. Assume that we've decided to change **Branch 7.x** after looking at

Where to commit.

1. Open Windows Explorer and right-click on an empty folder where you would like the development version to be downloaded into and select "SVN Checkout...".
   In this example, the folder is called `tikisvn`.

2. In the pop-up, type in the following as the URL of the repository: `http://svn.code.sf.net/p/tikiwiki/code/7.x`, as shown in this screenshot:



Checkout

Repository

URL of repository: https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki/branches/3.0

Checkout directory: C:\xampp\htdocs\tikisvn

Checkout Depth: Fully recursive

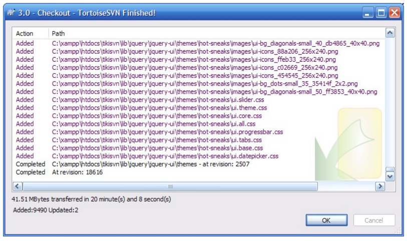
☐ Omit externals

Revision

● HEAD revision

○ Revision

3. After clicking OK, the pop-up will show the files being downloaded and will indicate when the download is complete as shown in this screenshot:



3.0 - Checkout - TortoiseSVN Finished!

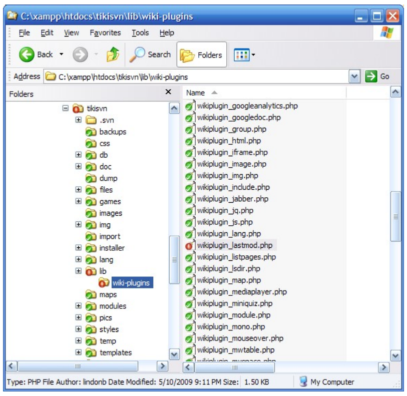
```
Action     Path
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\images\ui-bg_diagonals-small_40_db4865_40x40.png
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\images\ui-icons_88a206_256x240.png
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\images\ui-icons_ffeb33_256x240.png
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\images\ui-icons_c02669_256x240.png
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\images\ui-icons_454545_256x240.png
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\images\ui-bg_dots-small_35_35414f_2x2.png
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\images\ui-bg_diagonals-small_50_ff3853_40x40.png
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\ui.slider.css
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\ui.theme.css
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\ui.core.css
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\ui.all.css
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\ui.progressbar.css
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\ui.tabs.css
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\ui.base.css
Added      C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes\hot-sneaks\ui.datepicker.css
Completed  C:\xampp\htdocs\tikisvn\lib\jquery\jquery-ui\themes - at revision: 2507
Completed  At revision: 18616

41.51 MBytes transferred in 20 minute(s) and 8 second(s)
Added:9490 Updated:2
```

Step 2: Make Your Changes

1. Make changes to the file you wish to fix using a third party code editor.

   In our example, **lib/wiki-plugins/wikiplugin_lastmod.php** is being changed. After the change is saved, the changed status is indicated by the red color (the green color on the other folders and file icons show that the files were downloaded using the SVN program):



C:\xampp\htdocs\tikisvn\lib\wiki-plugins

Address: C:\xampp\htdocs\tikisvn\lib\wiki-plugins

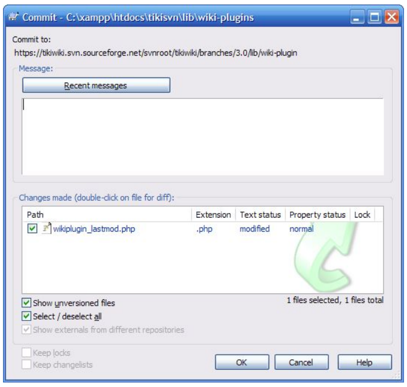
```
tikisvn
  .svn
  backups
  css
  db
  doc
  dump
  files
  games
  images
  img
  import
  installer
  lang
  lib
    wiki-plugins
  maps
  modules
  pics
  styles
  temp
  templates
```

```
wikiplugin_googleanalytics.php
wikiplugin_googledoc.php
wikiplugin_group.php
wikiplugin_html.php
wikiplugin_iframe.php
wikiplugin_image.php
wikiplugin_img.php
wikiplugin_include.php
wikiplugin_jabber.php
wikiplugin_jq.php
wikiplugin_js.php
wikiplugin_lang.php
wikiplugin_lastmod.php
wikiplugin_listpages.php
wikiplugin_lsdir.php
wikiplugin_map.php
wikiplugin_mediaplayer.php
wikiplugin_miniquiz.php
wikiplugin_module.php
wikiplugin_mono.php
wikiplugin_mouseover.php
wikiplugin_mwtable.php
```

Type: PHP File Author: lindonb Date Modified: 5/10/2009 9:11 PM Size: 1.50 KB

Step 3: Check Your Changes

1. In TortoiseSVN, this is done by right clicking the `tikisvn` folder and selecting "SVN Commit..." from the menu (even though you're not committing yet!). The following pop-up will appear:



Commit - C:\xampp\htdocs\tikisvn\lib\wiki-plugins

Commit to:
https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki/branches/3.0/lib/wiki-plugin

Message:

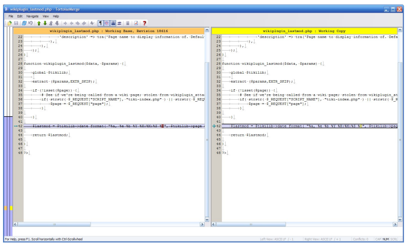
Changes made (double-click on file for diff):

Path	Extension	Text status	Property status	Lock
☑ wikiplugin_lastmod.php	.php	modified	normal	

1 files selected, 1 files total

☑ Show unversioned files
☑ Select / deselect all
☑ Show externals from different repositories
□ Keep locks
□ Keep changelists

2. Double click the file name shown in the above pop up to see the following record of changes made:



wikiplugin_lastmod.php - TortoiseMerge

wikiplugin_lastmod.php : Working Base, Revision 18616

```
                'description' => tra('Page name to display information of. Defaul
            ),
        ),
    );
}

function wikiplugin_lastmod($data, $params) {

    global $tikilib;

    extract ($params,EXTR_SKIP);

    if (!isset($page)) {
        # See if we're being called from a wiki page; stolen from wikiplugin_atta
        if( strstr( $_REQUEST["SCRIPT_NAME"], "tiki-index.php" ) || strstr( $_REQ
            $page = $_REQUEST["page"];
        }

    }

    $lastmod = $tikilib->date_format( "%a, %e %b %Y %H:%M:%S %Z", $tikilib->page

    return $lastmod;

}

?>
```

wikiplugin_lastmod.php : Working Copy

```
                'description' => tra('Page name to display information of. Defa
            ),
        ),
    );
}

function wikiplugin_lastmod($data, $params) {

    global $tikilib;

    extract ($params,EXTR_SKIP);

    if (!isset($page)) {
        # See if we're being called from a wiki page; stolen from wikiplugin_at
        if( strstr( $_REQUEST["SCRIPT_NAME"], "tiki-index.php" ) || strstr( $_R
            $page = $_REQUEST["page"];
        }

    }

    $lastmod = $tikilib->date_format( "%a, %e %b %Y %H:%M:%S %C", $tikilib->pag

    return $lastmod;

}

?>
```

For Help, press F1. Scroll horizontally with Ctrl-Scrollwheel

3. Check to see that these are the changes you intended

Step 4: Test Your Wiki Install

1. If you haven't already, install the developer version you have downloaded and changed by following the normal Installation procedures through an internet browser.

2. Once installed, test Tiki to ensure that your changes work properly.

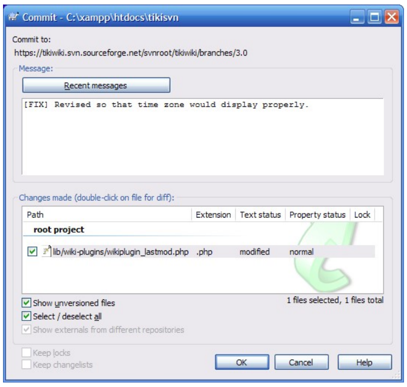
Step 5: Commit Your Changes

1. Type a description of the change you have made in the Commit pop-up that was brought up in the first part of Step 3 above and click OK. (If the pop-up is closed, right-click the `tikisvn` folder and select "SVN Commit..." from the menu to open it again.)

   The following tags can be used to distinguish changes:

   In the case of a "backport," (committing a change already made in trunk to an earlier version), a backport reference should be added in front. For example, `[bp/r28500] [FIX]...` in a commit message to the branch *branches/5x*, means this fix is a backport of trunk revision 28500 to the next version of Tiki 5x. Depending on the period, backports are subject to an extra level of review so please read Quality Team and Where to commit first or ask on the mailing list or IRC if unsure.

   Here's a screenshot showing the description filled in:



Commit - C:\xampp\htdocs\tikisvn

Commit to:
https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki/branches/3.0

Message:

```
[FIX] Revised so that time zone would display properly.
```

Changes made (double-click on file for diff):

Path	Extension	Text status	Property status	Lock
root project				
☑ lib/wiki-plugins/wikiplugin_lastmod.php	.php	modified	normal	

1 files selected, 1 files total

☑ Show unversioned files
☑ Select / deselect all
☑ Show externals from different repositories

□ Keep locks
□ Keep changelists

2. Enter your SourceForge username and password when prompted

3. The SVN program will notify you when the commit is completed and successful

Step 6: Be Proud of Yourself

Tiki is the collective work of hundreds of people. It works because of **volunteers**, like you, take the time to make it better.

Notice of your commit will appear in the Tiki IRC chat channel. Go here to see it in the logs or, better yet, log on to the actual chat channel just before you commit and see the notice pop up live!

alias

- Commit
- Contribute Code
- Share Code