View SVN source code:
https://sourceforge.net/p/tikiwiki/code/HEAD/tree/

Historical Revision Numbers:
See: https://sourceforge.net/p/tikiwiki/code/HEAD/tree/tags/ for the revision numbers of Tiki releases.

For descriptions about each branch, please see: Where to commit for more information.

Technical information

To get stable, development and experimental branches, see: Get code. Also see: Where to commit

First, the reference documentation for subversion:
http://svnbook.red-bean.com/

SVN client version

- You should use a recent version.
  - Starting in 10.x, we are using the "peg revision syntax" which requires SVN 1.5 (which is from June 2008)

If your SVN client is too old, you will get this error

```
Fetching external item into 'lib/test/^/third_party/vfsStream'
svn: Unrecognized URL scheme 'vfsStream'
```

    - If this is a big problem for you, we can revert
      - We cannot revert this one because of the difference between http and https on the new architecture. The SVN version on most of the provider is 1.5 which allow this kind of configuration. -changi

Basic SVN usage

Below are basic command lines to know. If you were using GUI like TortoiseCVS, just download TortoiseSVN instead ;)

- Tortoise SVN client: http://tortoisesvn.net/

Commands available to anyone on the internet

- To checkout the 19.x branch (creates a local SVN repository in your current directory including all the code):
  ```
  svn checkout https://svn.code.sf.net/p/tikiwiki/code/branches/19.x
  ```
- To checkout the development trunk (creates a local SVN repository in your current directory including all the code):
  ```
  svn checkout https://svn.code.sf.net/p/tikiwiki/code/trunk
  ```

- To build a new 19.x Tiki (all code without SVN specific files)

```
svn export https://svn.code.sf.net/p/tikiwiki/code/branches/19.x DEST_DIRECTORY
```

- To update a checkout (caution: current version of trunk is experimental)

```
cd checkout_directory
svn update
```

- To update/rollback an installation to a different revision (ahead or back)

```
svn update -r1234
```

- To update/rollback a single file to a different revision (ahead or back)

```
svn update -r1234 filename.php
```

- To update faster (in development, not production environments)

```
cd checkout_directory
svn up --ignore-externals
```

- To see what will be committed

```
cd checkout_directory
svn status
svn diff
```

- To list the content of 19.x branch on the server:

```
svn ls https://svn.code.sf.net/p/tikiwiki/code/branches/19.x
```

- To list the content of development trunk on the server:

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

Commands for Tiki developers

- **To commit something inside a checkout ie changing an existing file**

Committing requires a SourceForge login/password. Also, a Tiki admin needs to give you Write permissions. See How to get commit access

```
cd checkout_directory
svn commit
```

Please commit with an inline message:

```
svn commit -m "[FIX] short fix description...."
```

Your first commit may require your sourceforge login/password

```
svn commit --username yourlogin -m "[FIX] short fix description...."
```

- **Special considerations when adding a new file to the repository**

  Add a file

  ```
  cd checkout_directory
  svn add new_file
  svn commit -m "[ADD] feature: new file added in order to ....."
  ```

Please take care to add an Id keyword and to set the svn properties for this keyword for any new files that you add so that the revision/commit details are automatically added to the file whenever it is updated

- php, inc, tpl, js, css files should have only one svn property: svn:keywords whose value is "Id" (without double-quotes). Please, see this page for some background information on keywords. You should add, in the header of any new files a commented line:
  e.g. for a .php file it would look like this:

  ```
  // $Id$
  ```

  or for a .tpl file it would look like this:
  {* Id *}

Having added the keyword text to the file you then need to set the svn property which can be done with a svn propset command as follows:

```
svn propset svn:keywords "Id" yournewfile.php
```

Following which you should see the response:

```
property 'svn:keywords' set on 'yournewfile.php'
```

Once committed, the keyword will automatically expand and indicate the last commit on that file.

NB png, jpg, all image files : svn:mime-type ==> application/octet-stream. For these files, their prop is automatically added by the server when you commit

- **To see the log of commit messages**

  ```
  cd checkout_directory
  svn update #this is to be sure to have the last log entries
  svn log
  ```

  or page by page

  ```
  svn log |less
  ```

View one commit's changes

```
svn diff -r1234:1235
```

Rollback one version (e.g. revert changes made in a range of revisions)

```
svn merge -r1235:1234 .
svn ci
```

Alternative command to rollback one version (e.g. revert changes made in one revision)

```
svn merge -c -1235 .
svn ci
```

View single log entry

```
svn log -r1234
```

- **To compare folder/local files and SVN repository**

  ```
  svn status --show-updates
  ```

This will show you only the modified or added/missing files (not their content) between the folder you are in and the SVN repository.
Very useful before erasing your working in progress Tiki and you want to be sure you didn't omit to commit something.

Useful tips

Those commands are faster if you use them on one precise directory or file
(e.g. svn log tiki-index.php ; svn diff lib/ ; ...)

Your SourceForge login/password will not be necessary until you commit something.
This means that anonymous and developer access are the same... This is very nice 😃

Note that if you have a message like this one when trying to commit, it means that your password has expired and you need to login through SourceForge's web interface to change it. SourceForge will allow you to login and will not tell you the password is expired. Change it anyway.

```
svn: Commit failed (details follow):
svn: MKACTIVITY  of '/svnroot/tikiwiki/!svn/act/eac4ef53-cc7f-4415-ae1e-da1bac94a2ce':
 authorization failed (https://tikiwiki.svn.sourceforge.net)
```

Checksum problems

If you get a problem updating a working copy such as

```
svn: E155017: Checksum mismatch for '~/trunk/lib/wiki-plugins/wikiplugin_objecthits.php':
   expected:  82be35f524edc30a7bcef0b8d1350af2
   recorded:  06bb9a80cd27e4826ad392289a7cd193
```

you may find the tips on this page useful, especially for svn 1.7+

Checksum Issues 2019 Notes

This happened again in July 2019 (some sort of data storage issue in the repository) so here is the recipe that worked for most of us as posted on the dev list and based on the link above and other versions of it.

Note: This deletes all your files and replaces them with the latest versions in the repository. It takes several minutes so not ideal for production sites.

SVN Checksum Mismatch Fix

```
svn update --set-depth empty      # this deletes all your files
svn update --set-depth infinity   # this restores them but with "tree conflicts"
# quit tree conflict resolution
svn revert . --recursive          # and this fixes the conflict issues
```

MultiTiki

If you use MultiTiki and SVN together (for instance; for local development) then adding this line to your subversion config file

will help keep things tidy for you:

```
global-ignores = *.com *.org *.net *.local *.whatever
```

Important notice related to SourceForge migration

Some of the previous installation keep looking at the old repository of Tiki Wiki CMS Groupware at SourceForge (before mid December 2012). Please update (relocate) to the proper repo.

- If you want to checkout tiki code through svn while keeping the ability to commit from that installation when needed, you have to use the https paths, which will request your sf.net username and password:

  For 9.x LTS:

  command in a single line

  ```
  svn switch --relocate
  https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki/branches/9.x
  https://svn.code.sf.net/p/tikiwiki/code/branches/9.x ./
  ```

  You might be first asked the password for the same username in sf.net as the username you use on your server: just press intro with no answer. You will be then asked your sf.net username and password, and if you want to store plain text password (your are recommended not to store them as such): provide your credentials and choice, and you are done for the next step.

> *Authentication realm: SourceForge User*
> *Password for 'root':*
> *Authentication realm: SourceForge User*
> *Username: your.sf.net.user*
> *Password for 'your.sf.net.user': ********
>
> *~ATTENTION! Your password for authentication realm">*
>
> *SourceForge User*
>
> *can only be stored to disk unencrypted! You are advised to configure*
> *your system so that Subversion can store passwords encrypted, if*
> *possible. See the documentation for details.*
>
> *You can avoid future appearances of this warning by setting the value*

> *of the 'store-plaintext-passwords' option to either 'yes' or 'no' in '/home/your.server.user/.subversion/servers'.*
> *~*
> *Store password unencrypted (yes/no)? no*

+

command in a single line

```
svn up
```

+ Similarly, for 10.x: +

command in a single line

```
svn switch --relocate https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki/branches/10.x
https://svn.code.sf.net/p/tikiwiki/code/branches/10.x ./
svn up
```

+ and Similarly for mods: +

command in a single line

```
svn switch --relocate https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki/mods
https://svn.code.sf.net/p/tikiwiki/code/mods ./
svn up
```

- After you have successfully performed the previous action:
  If you want to checkout tiki code in that server through svn every now and then, but are not planning to commit from there, or not willing to add your sf.net credentials in that server to just checkout code, you can switch to use the non-https paths (which will not request your sf.net username and password):

  For 9.x LTS:

  command in a single line

  ```
  svn switch --relocate https://svn.code.sf.net/p/tikiwiki/code/branches/9.x
  http://svn.code.sf.net/p/tikiwiki/code/branches/9.x ./
  svn up
  ```

  For 10.x

  command in a single line

  ```
  svn switch --relocate https://svn.code.sf.net/p/tikiwiki/code/branches/10.x
  http://svn.code.sf.net/p/tikiwiki/code/branches/10.x ./
  svn up
  ```

If willing to checkout another branch, or trunk, update the paths accordingly. See below for the other paths if in doubt (trunk, mods, etc)

Right after switching and as you start to update you can encounter the following error that stops the update process:

```
svn: Delta source ended unexpectedly
```

To solve this the best I found is to find the file causing the problem and delete it.
Look at the last files that was updated, remove it and try again. If it didn't help, remove the all directory (if you modify directory content DO BACKUP) and try again.

Windows SVN Usage

Just because you don't know a grep from man page, doesn't mean you cannot contribute code. This section includes information for Windows users who prefer a GUI instead of a command-line.

Getting set up.

1. Get a Tiki and SourceForge account, then get SVN commit access.
2. Download and install a SVN client, such as TortoiseSVN, that you will use to checkin/out files.
3. Download and install an editor, such as Notepad++ or PhpStorm, that you will use to edit the TPL and PHP files.
   Note: When using your editor, whether you use a simple program like Notepad++ or something proprietary like Dreamweaver, you must make sure that you save your edited files with Unix-style line breaks (just LF instead of the CR + LF that Windows would normally use).

Checking out the code.

1. Download the Tiki source code to your PC.
2. Create an empty directory.
3. Right-click in the directory and select **SVN Checkout** from the popup menu.
4. In the Checkout dialog, enter the following:
   - **URL of respository**: https://svn.code.sf.net/p/tikiwiki/code/trunk or https://svn.code.sf.net/p/tikiwiki/code/branches/19.x
   - **Checkout directory**: *your local directory*
   - **Revision**: Select **HEAD Revision** to get the latest code
5. Click **OK**. TortoiseSVN will download all the files to your local folder.

Checking in files.

1. Right-click a file that you udpated, and select **SVN Commit** from the pop-up menu.
2. In the Enter Log Message dialog, enter the following:
   - **Message**: Enter a descriptive synopsis of your edits.
   - **File**: Select the file to check in.
     If you right-clicked a single file, only that file will be shown. If you right-clicked a folder, all files in that folder (and its sub-folders) that you have edited will be shown.
3. Click **OK**. TortoiseSVN will upload the file and commit it to the SVN repository.

Updating files.

1. Right-click a blank-area in your local folder, and select **SVN Update** from the pop-up menu.
2. TortoiseSVN will compare the files in your local folder with those in the SVN repository, and download the newer version, if available.

About the repository directories structure

Subversion is usually structured a bit differently (but better from my point of view) compared to CVS. There is generally three directories at the root level:

- **trunk/** => this is the development version (future 19.x at this date), where most of development is done. New functionalities are added here,
- **branches/X.x** (eg. branches/19.x) => stable branches repository.
- **branches/experimental** => repository for some experimental work before merging with trunk (once

almost cleaned),
- **tags/** => this is where we store a "snapshot" of each release. There should be no code modifications here.

We will have those "standard" directories + some others. There will be at least two additional directories:
- third_party/ => this contains one subfolder per third party library that is used by tiki but should not be modified (except to update it).
- mods/ => this contains the code of tiki mods, available on mods.tiki.org

So, this means that we will now have this directory structure inside our repository:

```
tikiwiki/
  trunk/
  branches/
    4.x/
    5.x/
    6.x/
    7.x/
    8.x/
    9.x/
    10.x/
    11.x/
    12.x/
    13.x/
   14.x/
   15.x/
   16.x/
   17.x/
   18.x/
   19.x/
    experimental/
      workspaces/
      webdav/
        .
        .
  tags/
    4.0/
    4.1/
     .
    5.0/
     .
  mods/
    trunk/
  third_party/
    adodb/
    ckeditor/
    smarty/
```

The real structure is here:
https://svn.code.sf.net/p/tikiwiki/code/

This means, for those who want to work on mods, that they will have to checkout the mods directory this way:

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

Restore a deleted file

1. Merge back specific commit from trunk

If you deleted a file in revision 19139

```
svn merge -r19139:19138 https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki/trunk
```

2. Commit file

Add details to explain why, and use right file name

```
svn commit -m "[FIX] Restoring file" filename.php
```

Restore in trunk a file deleted in a branch

1. Merge back specific commit from branches/19.x

If you deleted a file in revision 19139

```
svn merge -r19139:19138 https://svn.code.sf.net/p/tikiwiki/code/branches/19.x
```

2. Commit file

Add details to explain why, and use right file name

```
svn commit -m "[FIX] Restoring file" filename.php
```

How to figure out what revision number causes a bug

- How to figure out which commit causes a bug

What revision number am I at? What branch am I using?

svn info

```
$ svn info
Path: .
Working Copy Root Path: /var/www/htdocs/19.x
URL: https://svn.code.sf.net/p/tikiwiki/code/branches/19.x
Relative URL: ^/branches/19.x
Repository Root: https://svn.code.sf.net/p/tikiwiki/code
Repository UUID: b456876b-0849-0410-b77d-98878d47e9d5
Revision: 67730
Node Kind: directory
Schedule: normal
Last Changed Author: amnabilal
Last Changed Rev: 67727
Last Changed Date: 2018-09-29 07:45:37 +0300 (Sat, 29 Sep 2018)
```

Switching to another branch

For instance, if a checkout in *checkoutPath* is at version 18 and you want to upgrade to 19, do

```
svn switch https://svn.code.sf.net/p/tikiwiki/code/branches/19.x  checkoutPath
```

See Update

Important note for switching between 16.x and 17.x

The vendor directory is now in svn, so you will get an error `svn: Failed to add directory 'vendor': an unversioned directory of the same name already exists` if you switch from 16.x or earlier to 17.x or later (currently trunk).
This should be avoidable by deleting the vendor dir with `rm -rf vendor` HTH

If `svn status` shows :

```
D       vendor
D       vendor/.htaccess
```

then try :

```
svn -R revert ./
```

Other error may occurs like:

```
Fatal error: Uncaught exception
'Symfony\Component\DependencyInjection\Exception\InvalidArgumentException' with message
'You cannot set service "service_container".'
```

It is solved by deleting : temp/cache/container.php

Handling branches

Experimental branches

As of the release of Tiki2, releases will be made more often. In order to do so, trunk must remain somewhat stable. To make major changes, experimental branches can be used.

To work from an experimental branch:

1. Check-out trunk
2. Create your branch as follows:
    1. Open a Linux shell window (This is not tested on Windows)
    2. Position yourself at the root of the place where you checked out trunk.
    3. do `php doc/devtools/svnbranch.php branches/experimental/**your_subproject_name**`
        - Example: php doc/devtools/svnbranch.php branches/experimental/plugin_ui
3. Check-out your new branch
4. Develop and commit in your branch

1. When done and ready to merge, update one last time from latest trunk code
2. From the trunk check-out, run `php doc/devtools/svnmerge.php branches/experimental/**your_subproject_name**`
3. Commit changes with a meaningful message.

All scripts provide you with the next steps for each of them. Follow those steps carefully.

Update from Stable branches

- Semi-automatic merging period

Removing an experimental branch

Example of removing an experimental branch

```
svn remove -m "[KIL] Lesser magic was merged long ago"
https://svn.code.sf.net/p/tikiwiki/code/branches/experimental/lesser-magic/
```

Merging branch to trunk while releasing

Update from Stable branches

- Check-out trunk
- Run `php doc/devtools/svnbranchupdate.php branches/version_number`
  - Example: php doc/devtools/svnbranchupdate.php branches/10.x
- Fix conflicts and commit using `svn commit -F svn-commit.tmp`.

Notes from previous release processes

[+]
TODO

As you can see, I didn't create some directories yet:

- mods/branches/ : this is because we never had branches for mod before, so no urgency.

Another important thing: Do NOT touch libs of third_party, neither in third_party directory, nor in branches/x.x/lib/ or trunk/lib/ ... the two last ones are "specials" because they use svn capabilities (svn:externals property) to point to those in third_party directory. It's like links. They are there just to make it easier to have a running tiki with all needed libs from a checkout of trunk or a branch.

Merge a commit between branches

Please see: Merge a commit between branches

Related

- Subversion
- How to get commit access
- Where to commit
- Merge a commit to trunk
- Semi-automatic merging period
- Check if the update would cause conflicts

aliases

SVNTips | svn tip | svntip