

이 페이지의 번역은 완료되지 않았습니다.

티키 프로젝트는 [티키2](#) 에서부터 CVS 리포지토리에서 SVN (서브버전) 리포지터리로 리포지터리로 이전하였습니다.

소스 코드 보기:

<http://sourceforge.net/p/tikiwiki/code/HEAD/tree/>

기록 수정 번호:

참조: <http://sourceforge.net/p/tikiwiki/code/HEAD/tree/tags/> 에서 티키 배포판의 수정 번호를 참조하십시오.

각 브랜치에 관한 설명은, 다음을 참조하십시오: 추가 정보는 [어디에 커밋을 할 것인가](#) 를 살펴보십시오.

기술 정보

안정화, 개발용 혹은 실험용 브랜치들을 구하려면, 다음을 살펴보십시오: [코드 구하기](#). 추가 참조: [어디로 커밋을 할 것인가](#)

첫 째, 서브버전에 대한 참조 문서:

<http://svnbook.red-bean.com/>

SVN 클라이언트 버전

- 최근 버전을 사용하시는 것이 좋습니다.
 - 10.x 부터는 "peg revision syntax" (기존 수정 문법?) 를 사용하고 있으며, 이는 SVN 1.5 가 요구됩니다 (2008년 6월 부터)

귀하의 SVN 클라이언트가 너무 오래되었을 경우, 다음 오류를 받게 될 것입니다

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

- 이게 큰 문제가 된다면, 예전 방식으로 회귀도 가능합니다
 - 이것은 회귀가 불가능합니다, 그 이유는 새 아키텍처 상의 [http](#)와 [https](#) 사이의 차이때문입니다. 대부분의 제공자의 SVN 버전은 1.5이고 이는 이러한 종류의 구성을 허용합니다. `-changi`

기본 SVN 사용법

아래는 알아두어야할 기본 명령어입니다. TortoiseCVS 과 같은 GUI를 사용하는 경우, TortoiseSVN 를 대신 다운로드 하십시오 ;)

- Tortoise SVN 클라이언트: <http://tortoisesvn.net/>

인터넷상의 모든이에게 제공되는 명령어

- 12.x 브랜치를 체크아웃 하려면 (귀하의 현재 디렉터리 내부에 모든 코드를 포함하는 로컬 SVN 리포지터리를 생성하십시오):

```
svn checkout https://svn.code.sf.net/p/tikiwiki/code/branches/12.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 12.x Tiki (all code without SVN specific files)

```
svn export https://svn.code.sf.net/p/tikiwiki/code/branches/12.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 12.x branch on the server:

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

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...."
```

- 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

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

View single log entry

```
svn log -r1234
```

- Adding a 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 about the svn properties of the files that you add.

- 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](#). You can add, in the header of these files a commented line

```
// $Id$
```

Once committed, this will expand and indicate the last commit on that file

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

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: MKACTION of '/svnroot/tikiwiki!/svn/act/eac4ef53-cc7f-4415-ae1e-dalbac94a2ce':
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+

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 in 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
```


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 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++](#), [phpStorm](#) or [PHP Designer](#), 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 ie 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/6.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 updated, 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 within 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 13.x at this date), where most of development is done. New functionalities are added here,
- **branches/X.x** (eg. branches/12.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/  
    experimental/  
    workspaces/  
    webdav/  
    .  
    .  
  tags/  
    4.0/  
    4.1/  
    .  
    5.0/  
    .  
  mods/  
    trunk/  
  third_party/
```

```
adodb/  
ckeditor/  
smarty/
```

The real structure is here:

<http://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/4.x

If you deleted a file in revision 19139

```
svn merge -r19139:19138 https://svn.code.sf.net/p/tikiwiki/code/branches/4.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: .  
URL: https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki/branches/9.x  
--Repository Root: https://tikiwiki.svn.sourceforge.net/svnroot/tikiwiki--  
Repository Root: https://svn.code.sf.net/p/tikiwiki/code  
Repository UUID: b456876b-0849-0410-b77d-98878d47e9d5  
Revision: 42053
```

```
Node Kind: directory
Schedule: normal
Last Changed Author: marclaporte
Last Changed Rev: 42053
Last Changed Date: 2012-06-23 15:51:40 +0200 (Sa, 23 Jun 2012)
```

Switching to another branch

For instance, you are in 6 and you want to go into 9, do

```
svn switch https://svn.code.sf.net/p/tikiwiki/code/branches/9.x
```

See [Update](#)

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.

Related

- [Subversion](#)
- [How to get commit access](#)
- [Where to commit](#)
- [Merge a commit between branches](#)
- [Semi-automatic merging period](#)

aliases

[SVN](#) | [svn](#) | [svn](#)