**Using a GitHub *git* Repository as a Read-Only Mirror for an *svn* Repository**

**David T. Ashley (**[**dashley@gmail.com**](mailto:dashley@gmail.com)**)**

**Introduction and Overview**

I (Dave Ashley) use *svn* (Subversion) at home as a version control system. However, *git* and *GitHub* are very popular, and I wish to keep a copy of my public projects under *git* on *GitHub*. This requires an export from *svn* to *git*, as well as interaction with *GitHub*.

This document describes the procedures I use to initially export a complete *svn* repository to *git* and *GitHub*, and to incrementally export subsequent *svn* commits to *git* and *GitHub*.

***Git* vs. *TortoiseGit* *git* Executables**

At the time I first tried to fully export and *svn* repository to a *git* repository, I had both *Git* and *TortoiseGit* installed.

I discovered that the *git* binary included with *Git* seemed to handle the *svn* export incorrectly, eventually terminating with the error “*git-svn migration fatal: not a valid SHA1 …*”.

The *git* binary included with *TortoiseGit* seemed to handle the *svn* export correctly.

To use the *TortoiseGit* *git* binary from the *Bash* shell program distributed with *Git*, a longer path must be used. That is why *"C:/Program Files/Git/bin/git.exe"* rather than *“git”* is used in some of the instructions that follow. Using *“git”* alone would use the *git* executable distributed with *Git*, which does not do the export from *svn* correctly.

However, other than the export from *svn* to *git*, the *Git* *git* executable works correctly.

**Notes on My Home Setup**

I use *svn* hidden behind *SSH* port forwarding. For that reason, in the following procedures, the host for my *svn* repository is *locahost*. A typical *svn* URL in my home setup is “*svn://localhost/dtapublic*”.

**Initial *svn* to *git* Export**

. Create the file used to transform an *svn* author name to a *git* author name (if necessary). My file has this content and is named *authors-transform.txt*:

dashley = David T. Ashley <dashley@gmail.com>

. Perform the initial export:

$ "C:/Program Files/Git/bin/git.exe" svn clone svn://localhost/dtapublic -A authors-transform.txt

The transformation will proceed at the rate of approximately 20 *svn* revisions per minute. A subdirectory named *dtapublic* will be created to contain the *git* repository and sandbox. As the last phase, a *git* checkout will be performed.

. Create a *git* repository on *GitHub*.



*GitHub* also provides instructions about what can be done with the repository.



. *cd* into the new *git* repository (the subsequent *git* commands must be done from within the repository).

$ cd dtapublic/

. Execute the *git remote add* command below.

$ git remote add origin https://github.com/dtashley/dtapublic.git

. Execute the *git push* command below. This pushes the local repository to GitHub.

$ git push -u origin master  
Enumerating objects: 5953, done.  
Counting objects: 100% (5953/5953), done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (4931/4931), done.  
Writing objects: 100% (5953/5953), 63.54 MiB | 1.12 MiB/s, done.  
Total 5953 (delta 2765), reused 0 (delta 0)  
remote: Resolving deltas: 100% (2765/2765), done.  
remote:  
remote: Create a pull request for 'master' on GitHub by visiting:  
remote: https://github.com/dtashley/dtapublic/pull/new/master  
remote:  
To https://github.com/dtashley/dtapublic.git  
 \* [new branch] master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.

**Incremental *svn* to *git* Export**

. Run the git fetch command below.

$ "C:/Program Files/Git/bin/git.exe" svn fetch -A "C:/svnsbx/dtapublic/projs/dtats/trunk/projs/2018/20181116\_svn\_git\_mirror/authors-transform.txt"

A projs/dtats/trunk/projs/2018/20181116\_svn\_git\_mirror/authors-transform.txt

r255 = 98be7c63b17c13af3ac838588f1fee9281172560 (refs/remotes/git-svn)

. Right-click on the local *git* repository and choose *TortoiseGit -> View Log*. Ensure that “*All Branches*” is checked.



. The most recent commits should be shown at the top of the log. Right click on the topmost commit and choose *Reset “master” to this…*.



Choose “*hard*” reset to keep the *git* sandbox synchronized with the repository.



At this point, *origin/master* will be lagging *master*.



. Issue the *git push* command shown below. This will cause *origin/master* to become coincident with *master*.

dashley@DTA-T420-B MINGW64 /c/gitsbx/dtapublic (master)  
$ git push  
Enumerating objects: 15, done.  
Counting objects: 100% (15/15), done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (9/9), 827 bytes | 413.00 KiB/s, done.  
Total 9 (delta 2), reused 0 (delta 0)  
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.  
To https://github.com/dtashley/dtapublic.git  
 9425e85..98be7c6 master -> master

