SVN repository manual

Erik Hendriks
TUDelft, Faculty of Civil Engineering and Geosciences

August 26, 2019

The original version of this document was written by Giorgio Santinelli and Kees den Heijer of Deltares. It has been updated for the SANDBOX project, and serves as a guideline to get acquainted with SVN and SVN repositories. By no means does it provide an exhaustive overview of all the possible options that SVN can offer.

This document elaborates on working with Subversion, either via your web browser, TortoiseSVN or command line. For some suggestions about the structuring of a repository, the reader is referred to: `https://publicwiki.deltares.nl/display/OET/Raw+data+repository+structuring`.

1 SANDBOX repository Access

User accounts and passwords for the SADNBOX svn repository have been created and are created as follows:

- username: `firstnamelastname` e.g. chiucheng or erikhendriks
- password: svnsandbox

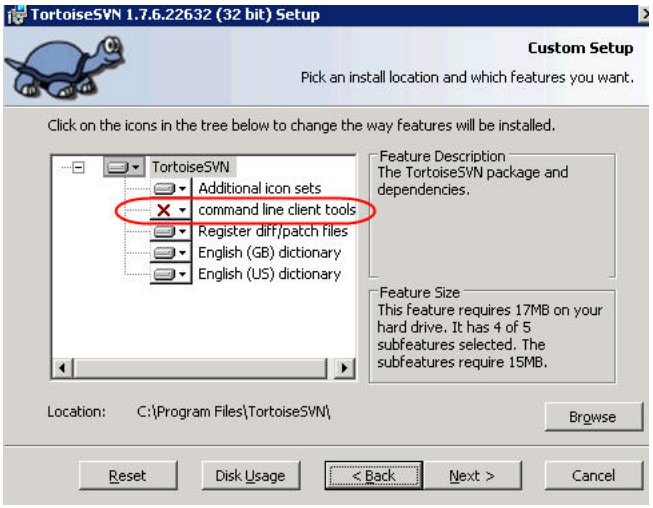
2 Browse

Go to the url of the repository, e.g. `https://svn.citg.tudelft/sandbox/trunk`. This allows you to browse through the directory structure of the repository and view individual files (text files only) or download them. TortoiseSVN includes a *repo-browser* that provides a similar overview of the directory structure and files.

3 Installation

To be able to also contribute to repositories, subversion client software needs to be installed on your machine.

- TortoiseSVN (`https://tortoisesvn.net/downloads.html`) is recommended for Windows.
  - It is recommended to enable the option *command line tools* during the installation of TortoiseSVN. By default the option is disabled, as indicated in the figure below.

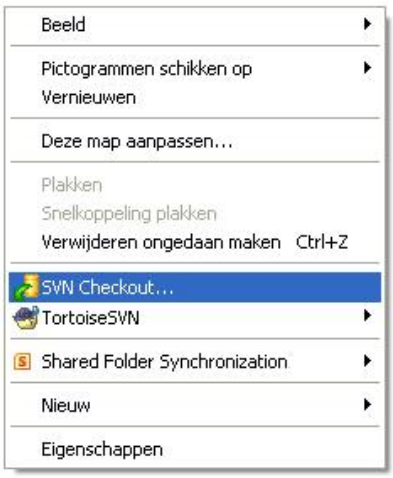


- SmartSVN (`https://www.smartsvn.com/download/`) is recommended for OS X, if you prefer to work with a user interface.
- Command line client tools are available for all platforms. On linux you can install it with:
  - `sudo apt install subversion` (Ubuntu/Debian)
  - `sudo yum install subversion` (Centos)

The right hand side of this tutorial gives some useful command line client commands.

4 Checkout

When making a *checkout*, one makes a local copy of the folders and files in the repository on his or her computer. A copy of the entire SANDBOX SVN repository will take up to 100 Gb of storage space (see Section 8), so making a copy of all the files and folders in the repository is not recommended. This section discusses several strategies for making checkouts. Making a partial checkout is discussed in more detail in Section 5.




```
svn checkout --username=<USERNAME>
https://svn.citg.tudelft/sandbox/trunk
```

PLEASE NOTE: this is the default checkout command which is certainly not recommended in case of large data repositories since the data volume will soon grow beyond the storage capacity of your computer.

- *Fully recursive:* Checkout the entire tree, including all child folders and sub-folders. *PLEASE NOTE: this is the default checkout command which is certainly not recommended in case of large data repositories since the data volume will soon grown beyond the storage capacity of your computer.*

```
svn checkout --username=<USERNAME>
--depth infinity
https://svn.citg.tudelft/sandbox/trunk
```

- *Immediate children, including folders:* Checkout the specified directory, including all files and child folders, but do not populate the child folders.

```
svn checkout --username=<USERNAME>
--depth immediates
https://svn.citg.tudelft/sandbox/trunk
```

- *Only file children:* Checkout the specified directory, including all files but do not checkout any child folders.

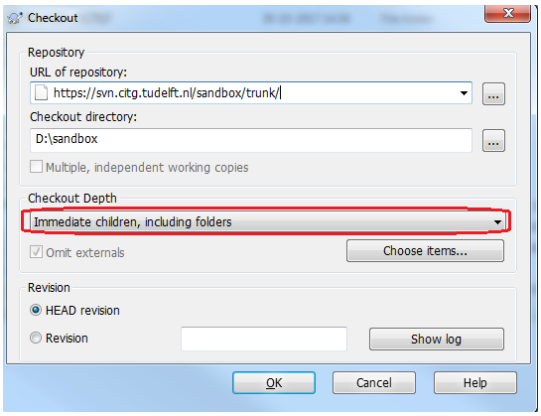
```
svn checkout --username=<USERNAME>
--depth files
https://svn.citg.tudelft/sandbox/trunk
```

- *Only this item:* Checkout the directory only. Do not populate it with files or child folders.

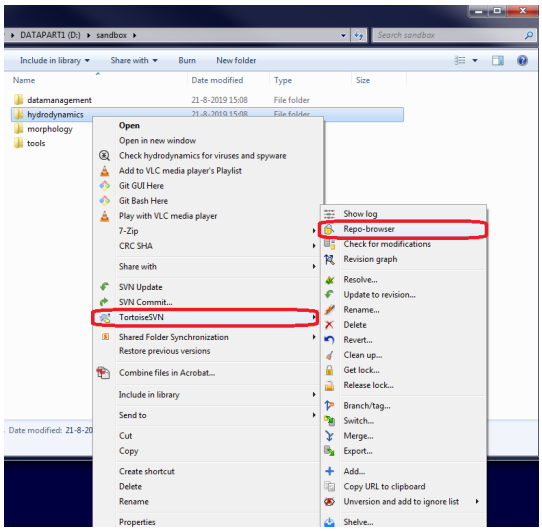
```
svn checkout --username=<USERNAME>
--depth empty
https://svn.citg.tudelft/sandbox/trunk
```

5 Partial checkout

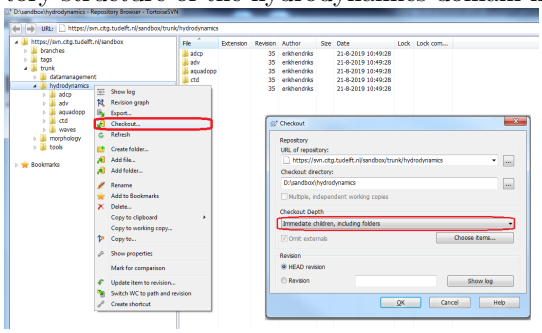
The recommended working procedure for making a *partial checkout* of the SANDBOX repository is discussed in this section. First, a checkout of the main folder (*trunk*) must be made, with checkout depth set to *'Immediate children, including folders'*, see the figure below.



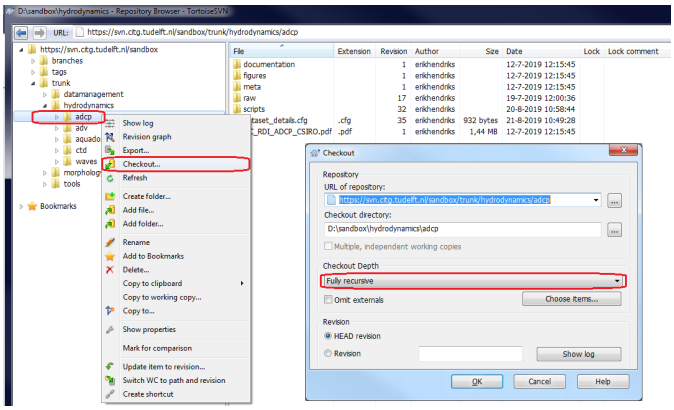
If SVN Tortoise then asks for your username and password, key in your personal username and password as specified in Section 1. If this is done successfully, SVN Tortoise makes a checkout of the main directory structure. Currently, there are 4 main domains (i.e., folders) in the repository: datamanagement, hydrodynamics, morphology and tools. We will demonstrate how to make a partial checkout of the *adcp* dataset from the *hydrodynamics* domain. First, right-click the hydrodynamics folder in the local checkout directory. Then click the *repo-browser* option in the SVN Tortoise submenu.



In the repo-browser, right-click the hydrodynamics subfolder and choose for the option *Checkout*. Again, make sure that in the Checkout pop-up, the Checkout Depth is set to *'Immediate children, including folders'*. This creates the directory structure of the hydrodynamics domain in your local checkout folder.



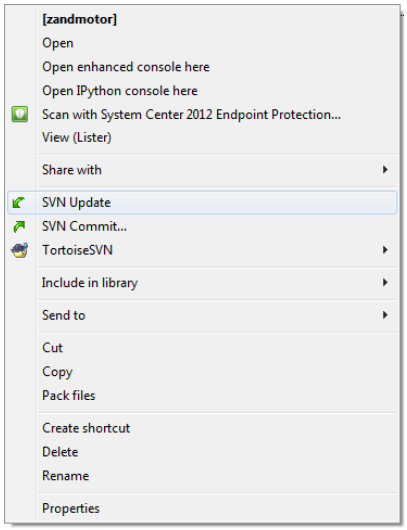
Once the directory structure is downloaded, we proceed by downloading the *adcp* dataset. This is done similarly to how the directory structure for the hydrodynamics domain was downloaded. However, we now set Checkout depth to *'Fully recursive'*. See the figure below for this final step. This procedure can be repeated for the different datasets in the 4 main domains.



6 Important Actions

The two most important commands are *SVN Update* and *SVN Commit*. It is very important to use *SVN Update* regularly, but especially when you start working on a certain file/directory. If this command is not used, it might happen that someone else in the meantime has been updating the file/directory; which means that you are working with an older version of this file/directory. On the other hand, it is very important to use *SVN Commit* at the end of the day. This command uploads the files you have been working on, so all others will be able to work with the most recent versions. When you use *SVN Commit*, you should always write a log message in English. This message gives everyone the possibility to see what is changed. You can either commit one file or folder, or the entire tree.

6.1 Update



Assuming that you are in the path of your checkout, use:

```
svn up
```

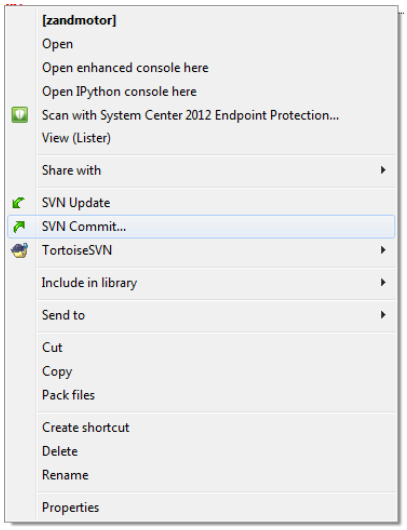
or

```
svn update
```

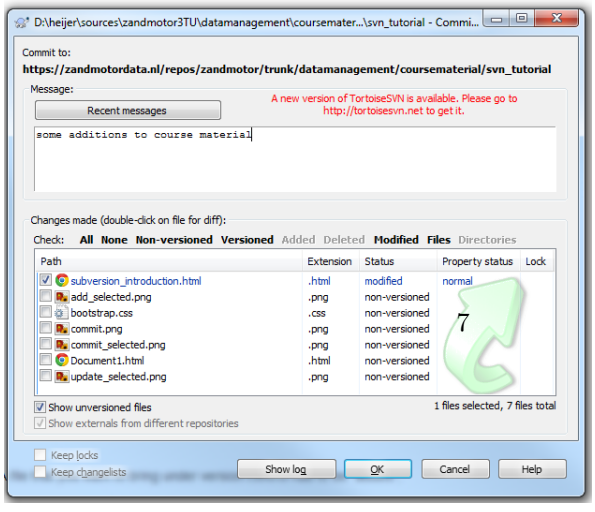
6.2 Commit

By committing, you are performing the actual uploading of your local modifications, additions or deletions to the server.

Right-click on the file or folder that you want to commit and select *SVN Commit* from the list.



A window pops up that provides you with an overview of the items that you are about to commit. Make sure that you fill in a useful but concise message in the message window (in English). If you are doing similar commits multiple times, it is useful to make use of the *Recent messages* to save the number of key strokes. Only the files that are selected (tick box left) will be send to the server.



To get an overview of the files that you are about to commit, use:

```
svn st
```

or

```
svn status
```

The actual commit action is performed by the following command:

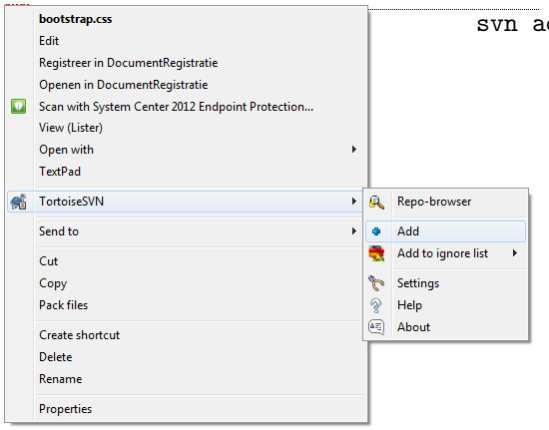
```
svn ci -m"write your custom concise
summary of your contribution here"
```

Make sure that you provide a useful but concise message between the quotes (in English).

Ticking the box of a *non-versioned* file has the same effect as *adding* a file.

6.3 Add

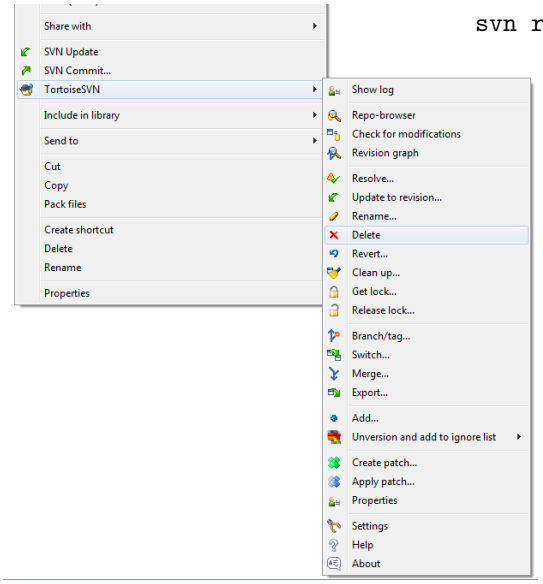
A newly created file in your local file system is by default not under version control. A file that you want to bring under version control has to be *added*, meaning that you nominate the file to be uploaded on the next commit.




```
svn add FILENAME
```

6.4 Delete

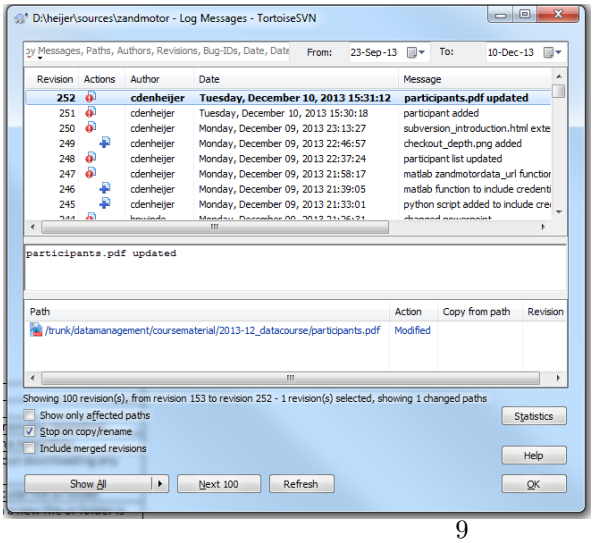
Locally deleting a file does not automatically mean that it is also deleted on the server. If the server does not know that the file should be deleted, it will see the file as locally missing, and will restore the file on the next update. By using the *Delete* option of your SVN client, the file will be deleted locally and remember that it has to be deleted on the server during the next commit action.




```
svn rm FILENAME
```

6.5 Log

To get an overview of the activity on the server, the SVN log can be used.



The following command provides you with all the log messages, which can be a long list:

```
svn log
```

To limit the number of records, you can limit them to e.g. 10 like this:

```
svn log -1 10
```

7 SVN Keywords

SVN keywords are used to include parameters like author, version and date automatically in your plain text files (e.g. Python, Matlab, LaTeX). If the specific tags appear in files, Subversion is able to fill in and maintain this information. These tags are:

- `Date`
- `Revision`
- `Author`
- `HeadURL`
- `Id`

When these tags are between `$ $` signs *AND* the file has the corresponding properties set, then subversion will complete the information as follows:

```
$Date: 2018-01-26 14:25:17 +0100 (vr, 26 jan 2018) $
$Revision: 14113 $
$Author: heijer $
$HeadURL: https://svn.oss.deltares.nl/repos/openearthtools/trunk/latex/svn_manual/contents/t
$Id: tutorial_svn_keywords.tex 14113 2018-01-26 13:25:17Z heijer $
```

See also `https://wiki.documentfoundation.org/Svn:keywords` for more information. Setting the related properties only needs to be done once for each file. It can be done in two ways:

automatic Properties of newly created files with particular extensions will automatically be set (see Section 7.1).

manual Properties of new or existing files can be set or modified (see Section 7.2).

7.1 Enable automatic properties in the config-file

First find the config file and open it with a text editor. The location of the file varies across different platforms.

Windows

- Command line:

  ```
  set
  ```

- Search for the location of APPDATA and go to the Subversion folder; this folder is:

  ```
  C:\Users\<USERNAME>\Appdata\Roaming\Subversion
  ```

- Open *config* with a text editor

OSx / Linux

- Open the config-file with a text editor. For example with `vi` on the command line:

  ```
  vi ~/.subversion/config
  ```

7.1.1 Adjust the config-file

- Remove the `#` in front of `enable-auto-props = yes` to enable this property
- Create:

  ```
  *.py = svn:keywords=Author Date Id Rev URL
  *.m = svn:keywords=Author Date Id Rev URL
  *.tex = svn:keywords=Author Date Id Rev URL
  *.txt = svn:keywords=Author Date Id Rev URL
  *.R = svn:keywords=Author Date Id Rev URL
  ```

7.2 Manually set svn:keywords property to a folder or file

7.2.1 Windows

- Go to the folder or file, you want to modify, in your checkout
- Right click and choose *TortoiseSVN - properties*
- Select *new - keywords* and select the keywords you want

7.2.2 Command line

In command line:

```
svn propset svn:keywords "Author Date Id Rev URL" folder/file location
```

8 SANDBOX SVN repository overview

date August 26, 2019

dataset	domain	date	contact	volume	readme	scripts	thredds
adcp	trunk/hydrodynamics	2019-08-20		828.2 MB		.m, .py	
adv	trunk/hydrodynamics	2019-07-19		7.8 GB		.py	
aquadopp	trunk/hydrodynamics	2019-07-19		6.3 GB		.m, .py	
ctd	trunk/hydrodynamics	2019-08-21		1.1 GB			
waves	trunk/hydrodynamics	2019-07-12		24.6 MB		.py	
adv_obs	trunk/morphology	2019-08-19		4.5 GB		.py	
aquascat	trunk/morphology	2019-08-19		658.0 MB			
floc_camera	trunk/morphology	2019-07-15		11.4 GB			
lisst100x	trunk/morphology	2019-08-19		872.2 MB		.m, .py	
lisst200x	trunk/morphology	2019-08-19		134.8 MB			
multibeam	trunk/morphology	2019-08-19		2.8 GB			
obs	trunk/morphology	2019-08-19		83.1 MB			
rippleprofiler	trunk/morphology	2019-08-20		11.9 GB		.m, .py	
sediment_multicore	trunk/morphology	2019-08-19		167.6 KB			
sediment_spi	trunk/morphology	2019-08-19		3.2 GB			