# TortoiseSVN

## A Subversion client for Windows

## Version 1.4.5

# Installing TortoiseSVN

Click here to download the latest version of the Installer of 32bit TortoiseSVN

Or

Copy/paste the following link to your web-browser: http://tortoisesvn.net/downloads

Once you have downloaded the TortoiseSVN client, you need to install this client on your system. Here are the steps to install TortoiseSVN client:

- Locate the path where you have downloaded TortoiseSVN installer.
- After the installation from the .msi file, you can start using TortoiseSVN. But a restart is recommended.

# TortoiseSVN User Guide

Let us now learn to execute some basic Subversion features using TortoiseSVN. Once you install TortoiseSVN, you will find new options added to the Windows right click popup menu. The actions associated with TortoiseSVN will be available at the right click as shown below:
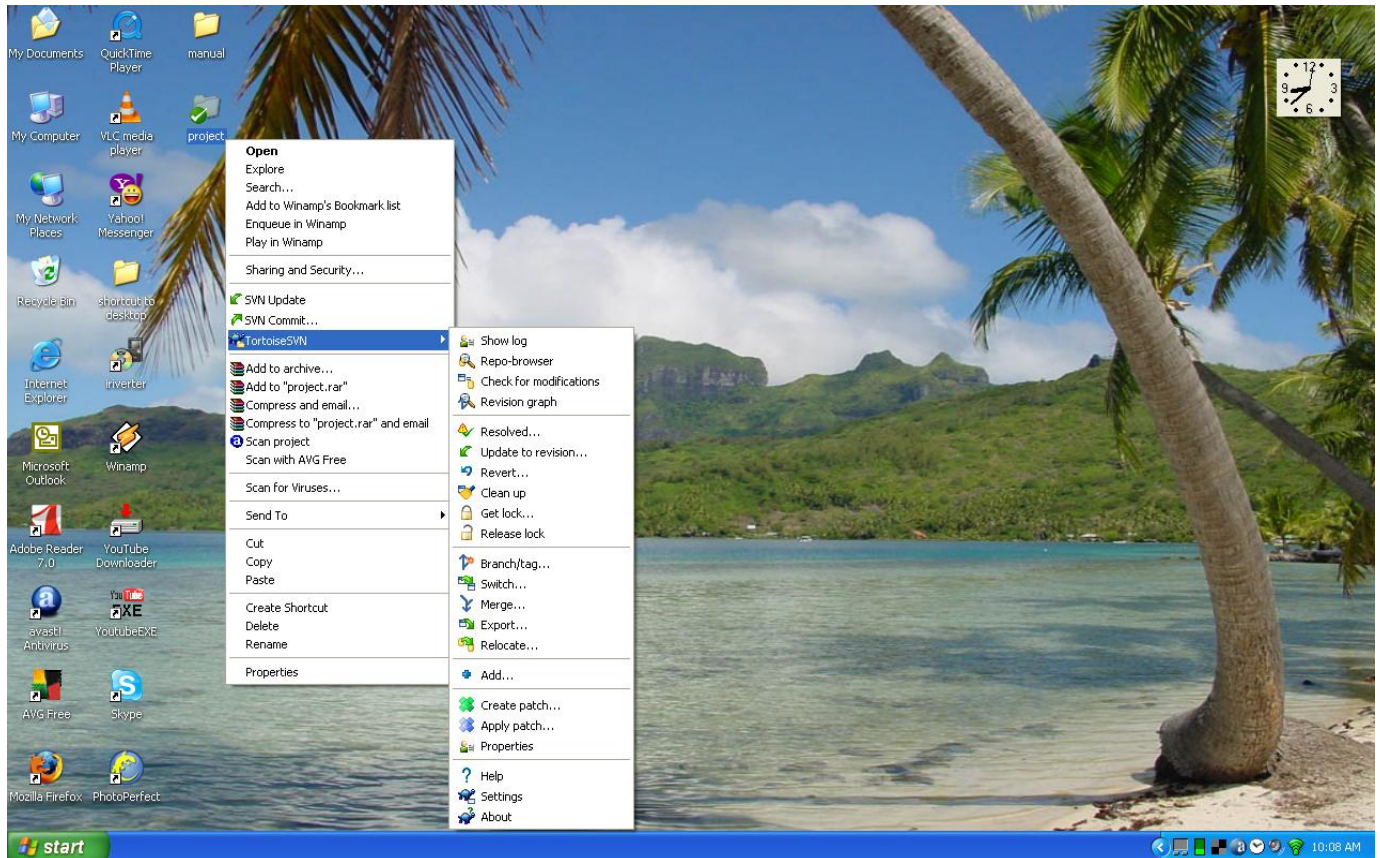


Repository: A repository is a place where data is stored and maintained.Usually it is stored on the main server which contains data of all the projects. A client/user requires to access the repository to perform various actions like add/modify/delete etc. with the contents of their respective project.

Working Copy: To use this Subversion client, or any other subversion client, we first need a local copy in our system of the repository contents. This local copy is also known as the 'Working copy'. This is where we make all modifications and then send these changes to the repository.

# Downloading a Repository to a Working Copy (SVN Checkout)

1. Create a folder in your system where you want to download (checkout) the repository contents.
2. Move your mouse to point at this folder, and do a right click
3. Select the 'SVN Checkout' option
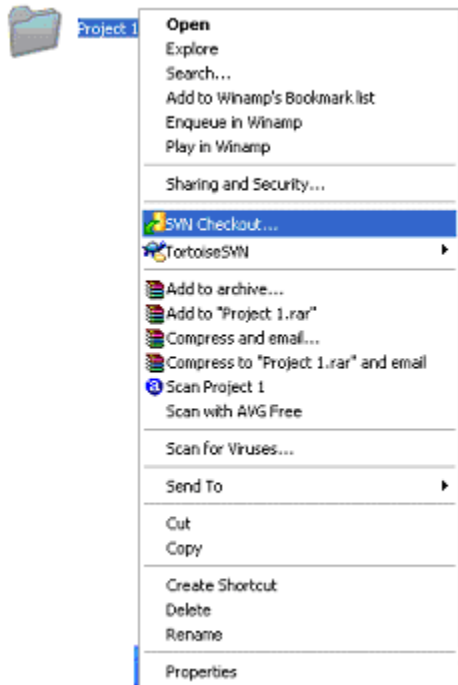


4. A dialogue box as shown below opens up.
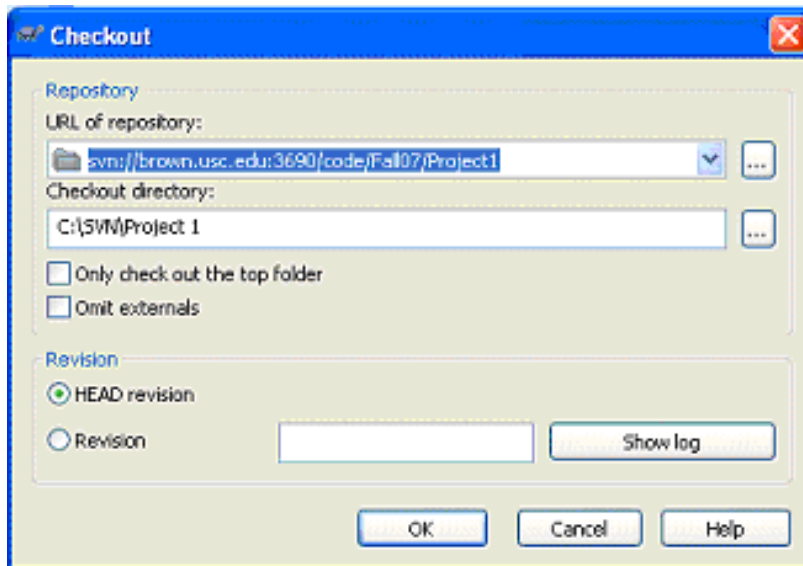
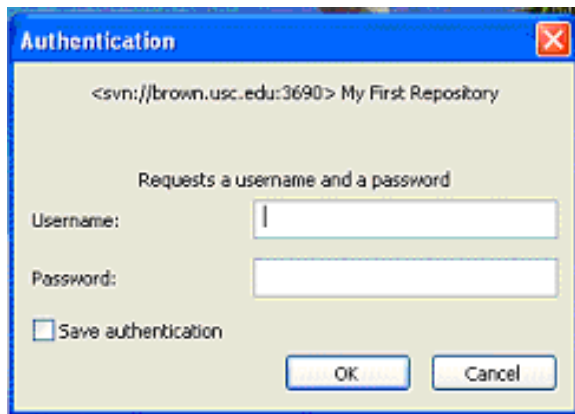This dialogue box requires you to enter the following details

- URL of repository: Enter the repository URL here.

svn://brown.usc.edu:3690/cs577a08/Team <#>

where Team <#> is the team number. Note the space between Team and <#>

For example, URL for team 1 is "svn://brown.usc.edu:3690/cs577a08/Team 1"

5. Authorization:



6. Enter your username and password in the authentication window, and click ok. Check the Save authentication box to avoid repetitive authorization.

7. Once username and password are authenticated, the repository files will be checked out into the working copy and a window like the one below will appear, and then click ok.
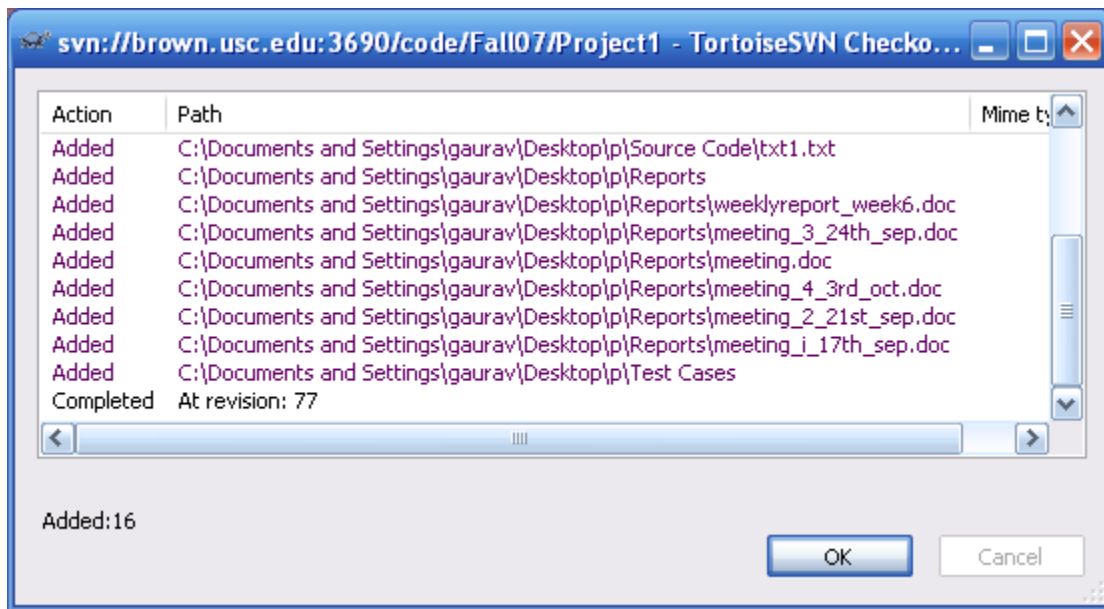
8. When you open the working copy folder you will see icons with a green tick mark sign. If you encounter any other sign you can refer them by icon overlays mentioned in the end.
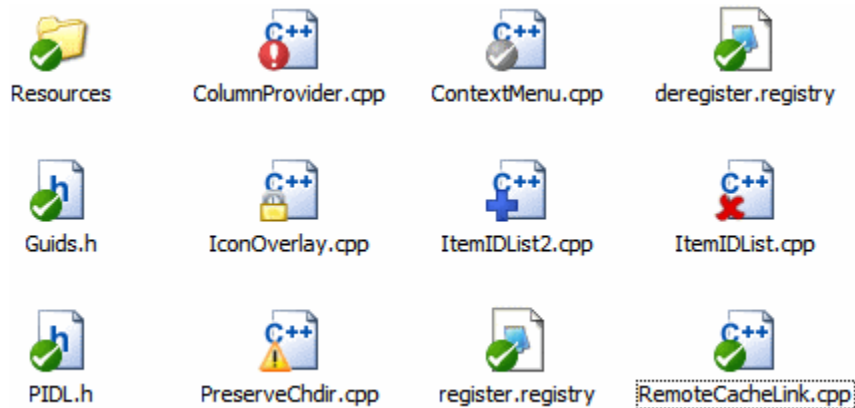
## **Update the Working Copy**

To update the working copy with the latest repository modification we use the SVN Update command. It merges changes in repository with the working copy.

1.  Right click on the folder/file
2.  Select 'SVN Update'
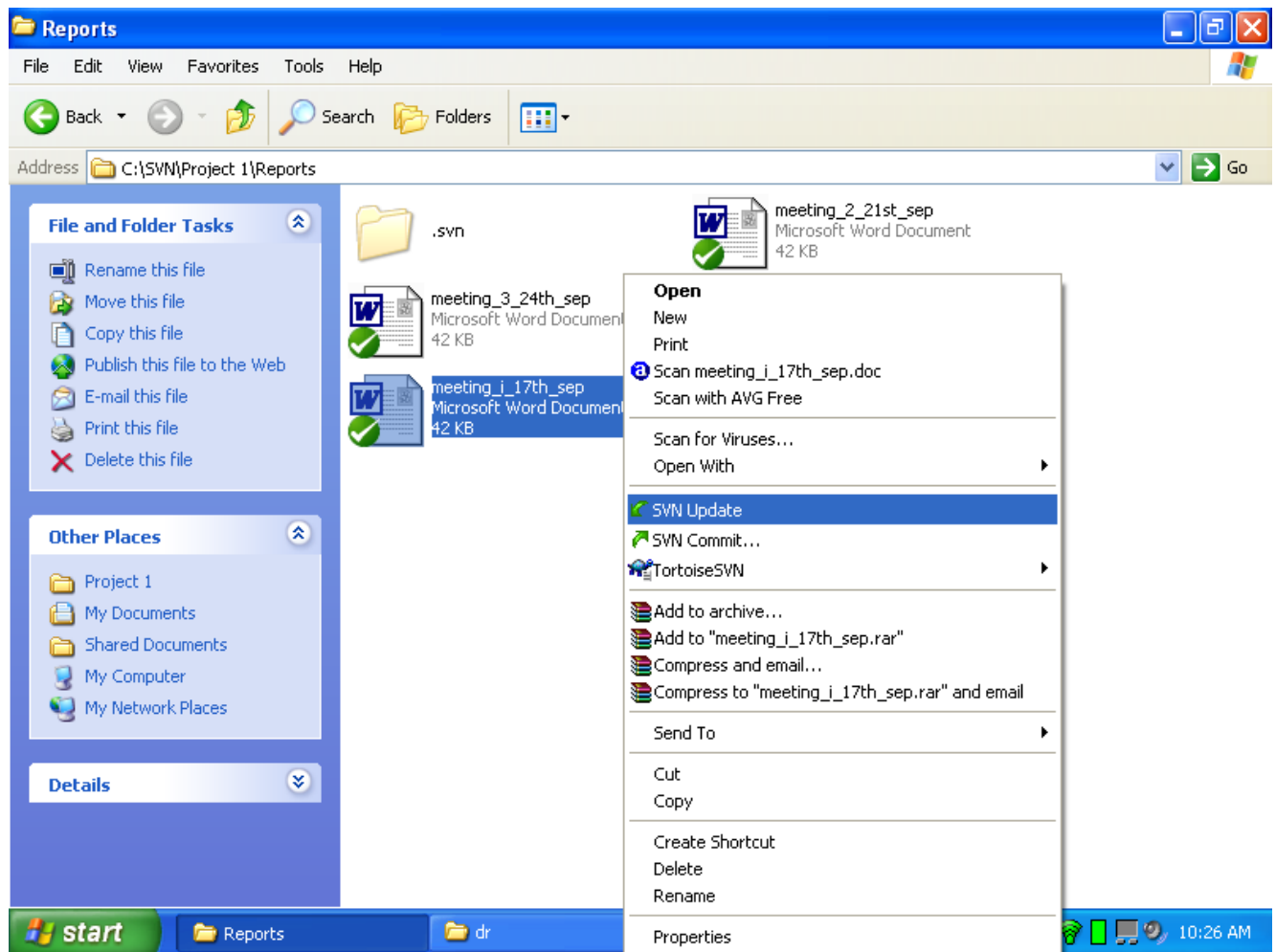
3. Click 'OK'

# Icon Overlays



Now that you have checked out a working copy from a Subversion repository you can see your files in the windows explorer with changed icons. TortoiseSVN adds a so called overlay icon to each file icon which overlaps the original file icon. Depending on the Subversion status of the file the overlay icon is different.

A fresh checked out working copy has a green checkmark as overlay. That means the Subversion Status is normal.

As soon as you start editing a file, the status changes to modified and the icon overlay then Change to a red exclamation mark. That way you can easily see which files were changed since you last updated your working copy and need to be committed.

If during an update a conflict occurs then the icon changes to a yellow exclamation mark.

If you have set the svn: needs-lock property on a file, Subversion makes that file read-only until You get a lock on that file. Read-only files have this overlay to indicate that you have to get a lock first before you can edit that file.

If you hold a lock on a file, and the Subversion status is normal, this icon overlay reminds you that you should release the lock if you are not using it to allow others to commit their changes to the file.

This icon shows you that some files or folders inside the current folder have been scheduled to be deleted from version control or a file under version control is missing in a folder.

The plus sign tells you that a file or folder has been scheduled to be added to version control. Unlike TortoiseCVS (the CVS shell integration) no overlay icon for unversioned files is shown. We do this because the number of icon overlays are limited system wide and should be used economically.

## Modify contents in the Subversion repository

1. Make the required modifications in the file/folder of your working copy.
2. A red exclamation overlay will appear on the modified file/folder.
3. Right click on the file that was modified.
4. Select the option 'SVN Commit'.
5. Enter a message.
6. Click 'OK'.
7. The red exclamation overlay on the modified file/folder will convert to a green tick mark overlay.

## Add a new file/folder to Subversion repository

1. In your working copy navigate to the folder where you want to add the new file/folder.
2. Add a new file/folder, for example meeting_7_21$^{st}$_oct.doc
3. Right click on the file meeting_7_21$^{st}$_oct.doc
4. Select TortoiseSVN - >Add
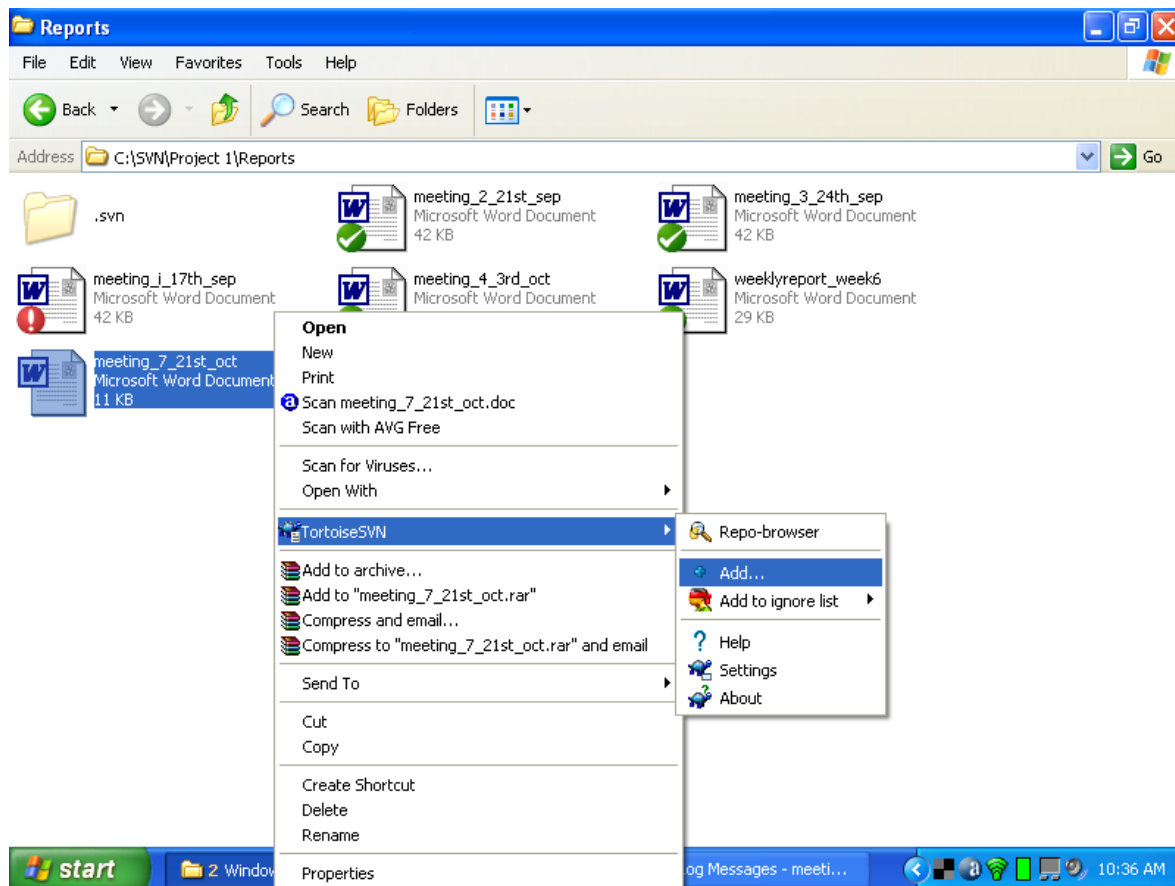
5. The 'TortoiseSVN: Add' box will be displayed which will confirm the file that is to be added. Click 'OK'
6. The 'TortoiseSVN Add ... Finished!' box will be displayed with a success message. Click 'OK'
7. A blue plus overlay appears on the added file/folder.
8. To send this change from the working copy to the repository, you need to perform a 'Commit' operation. Right click on the file meeting_7_21$^{st}$_oct.doc
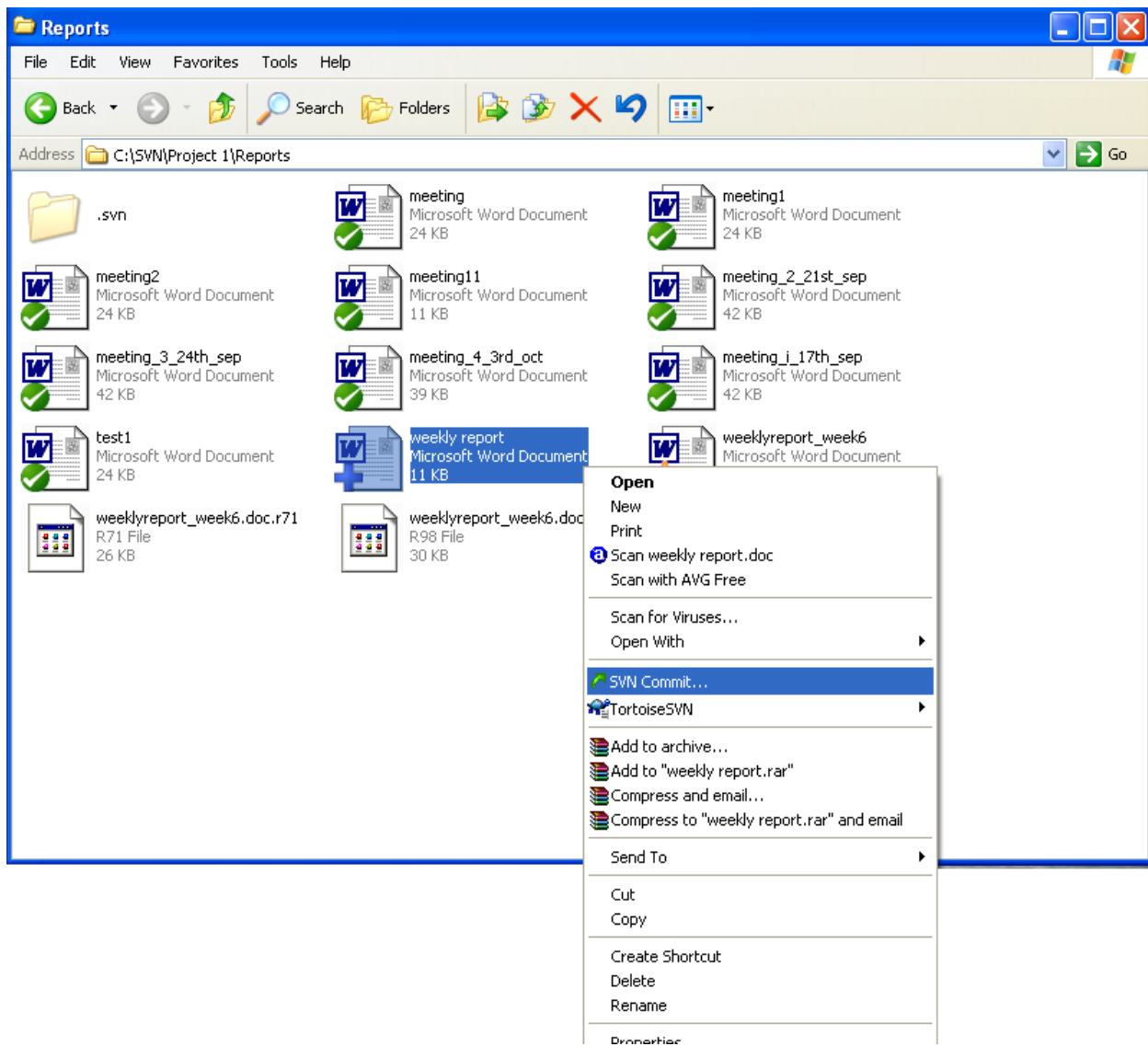
9. Select the option 'SVN Commit'
10. Enter a log message
11. Click 'OK'
12. The blue plus overlay on the added file/folder will turn to green tick mark.

NOTE:
The steps to add file or a folder or a folder tree are exactly the same.
When you add a folder tree, it's enough you perform the above steps on the parent folder.
The sub folders and files present in this folder are added and committed together with the parent.

# Delete a file/folder

1.  Right click on the file/folder that you want to delete
2.  Select TortoiseSVN - >Delete

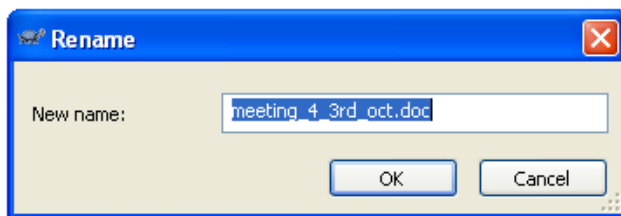3.  A red exclamation overlay will appear on the deleted file/folder's parent folder.
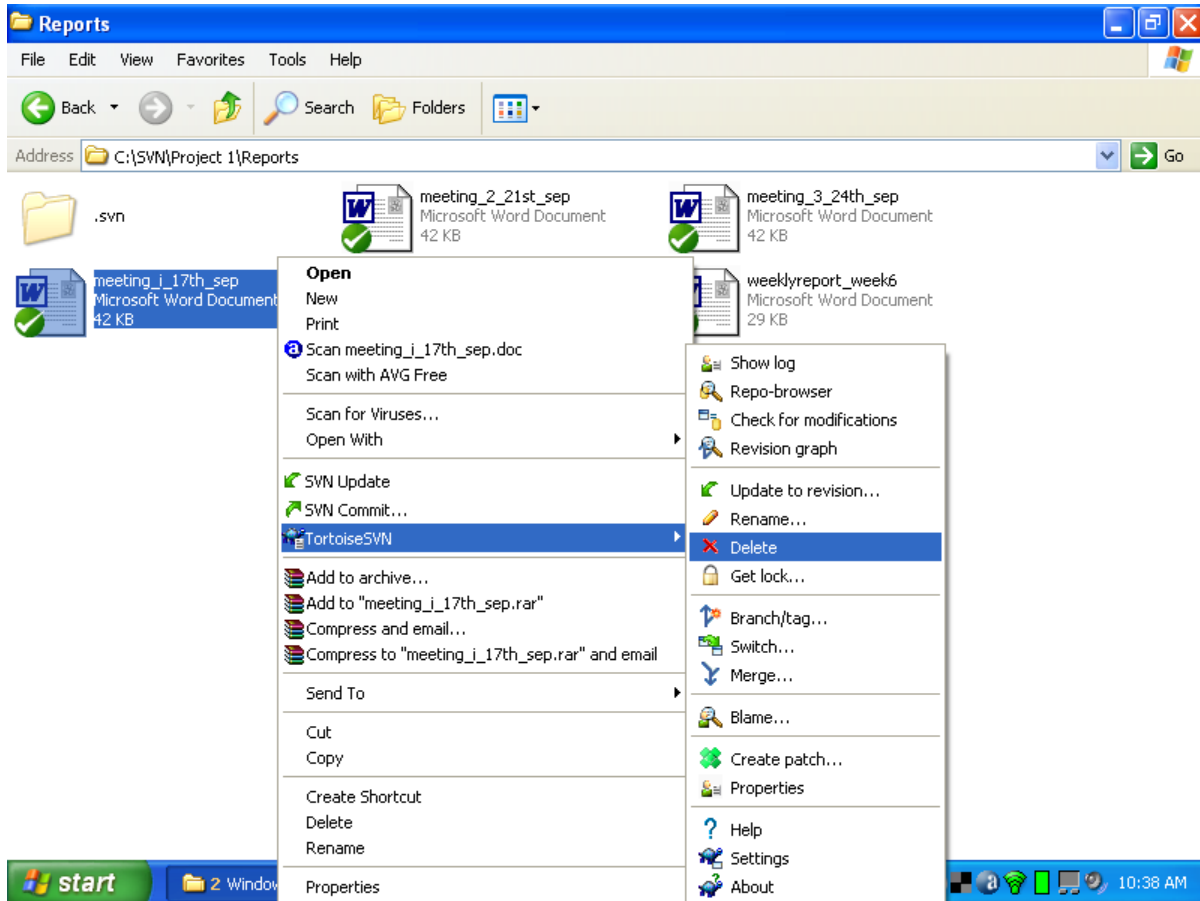


4.  Navigate to the parent folder and right click on it
5.  Select the option 'SVN Commit'
6.  Enter a message
7.  Click 'OK'
8.  The red exclamation overlay from the parent folder will convert to a green tick mark overlay.

# Rename a file/folder

1. Right click on the file/folder that you want to rename
2. Select the TortoiseSVN - >Rename
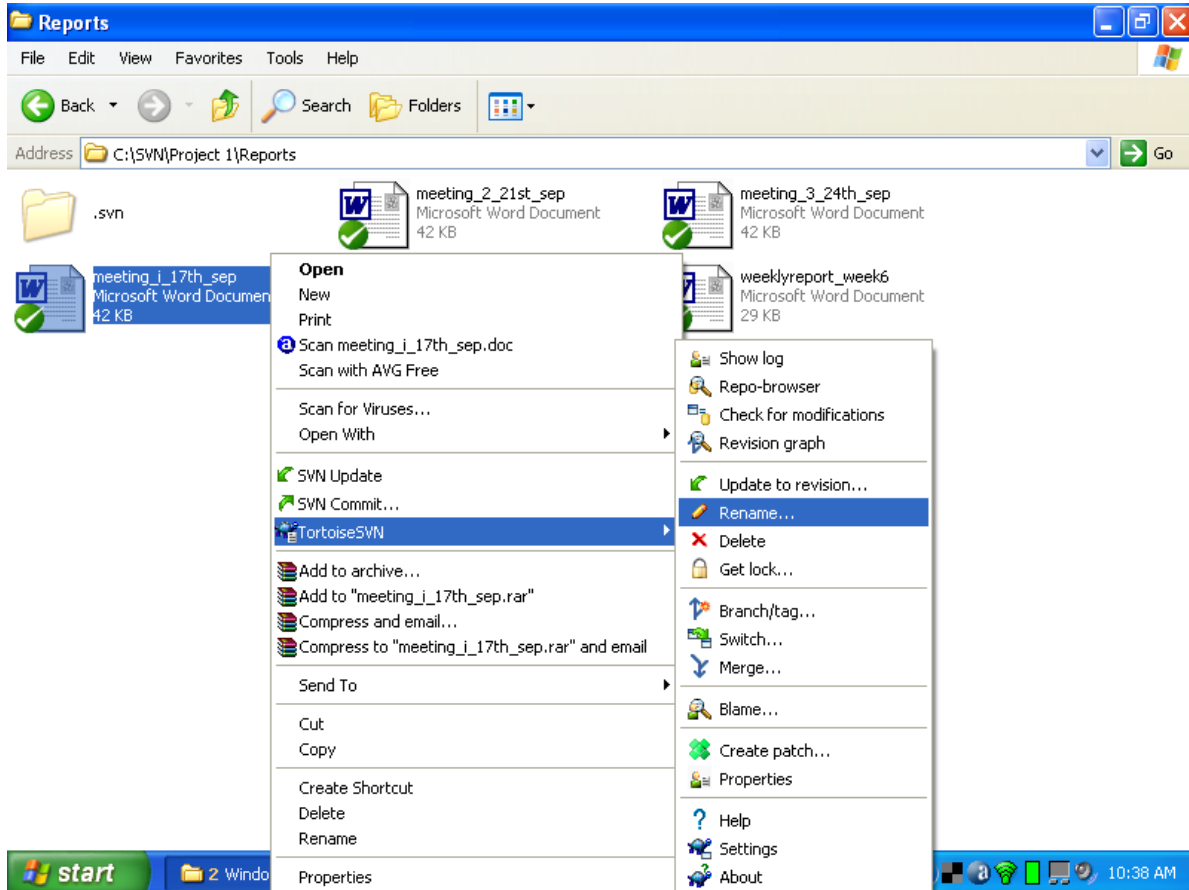


3. A pop window will open wherein you need to enter the new name for the file or folder



4. A red exclamation overlay will appear on the renamed file/folder's parent folder.
5. Navigate to the parent folder and right click on it
6. Select the option 'SVN Commit'
7. Enter a log message
8. Click 'OK'
9. The red exclamation overlay from the parent folder will convert to a green tick mark overlay.

# Revert Changes

If you wish to remove all changes made to the working copy that have been added but not committed, you can use the revert operation. This will undo all changes made to a file since the last commit.

1. Make a few changes to the contents of a file.
2. A red exclamation overlay will appear on the modified file/folder.
3. Right click on the file/folder of which you want to revert the changes
4. Select the TortoiseSVN - >Revert
5. The red exclamation overlay on the modified file will convert to a green tick mark overlay.
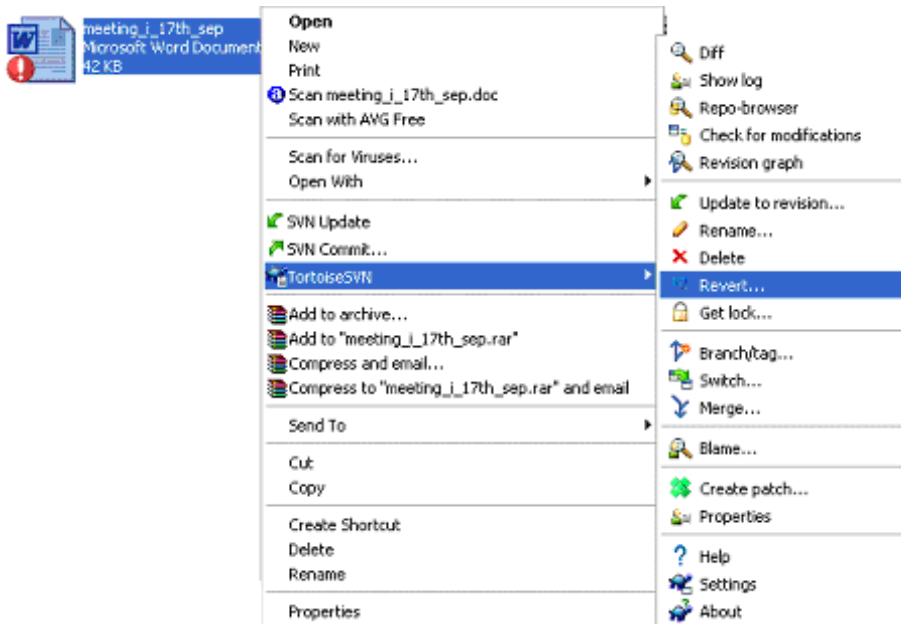
# Copy a file/folder

This command is used to copy a file/folder from your working copy to your system

1.  Right click on the file or folder that you wish to copy
2.  Without releasing the mouse button drag the file/folder to its destination
    **NOTE:** Use the right click to drag
3.  Drop the file/folder at the destination folder and you'd see a pop up menu like the following:
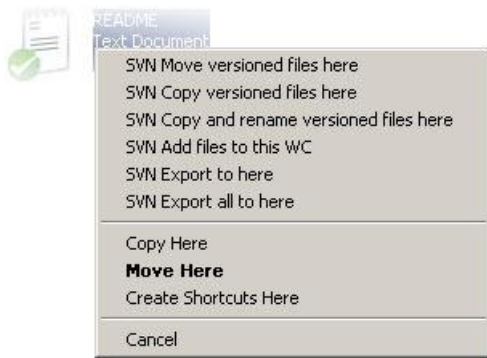


4.  Select one of the options

*   'SVN Move versioned files here' : Moves the file/folder from one location to another without loosing its history
*   'SVN Copy versioned files here' : Copies the file/folder from one location to another without loosing its history
*   'SVN Copy and rename versioned files here' : Copies and Renames the file/folder from one location to another without loosing its history
*   'SVN Add files to this WC' : Adds the file/folder as a new file to the Working copy
*   'SVN Export to here' : Exports the file/folder to the repository
*   'SVN Export all to here' : Exports all the files and folders in the source directory to the repository
*   'Copy here' : Copies the file/folder from source to destination as a new file
*   'Move here' : Moves the file/folder from source to destination as a new file
*   'Create shortcuts here' : Create a shortcut in destination folder
*   'Cancel' : Copy/Move operation is cancelled

5.  For copying, select 'SVN Copy versioned files here'
6.  Navigate to the parent folder and right click
7.  Right click on the file/folder
8.  Select the TortoiseSVN - >Commit option
9.  Enter a log message
10. Click                                                                            'OK'

# <u>View Log</u>

You can view the various log messages entered for a particular file/folder during commit operations.

1. Right click on the file/folder
2. Select TortoiseSVN - >Show Log
3. The 'Log Message' window will display the various log messages entered for a file/folder
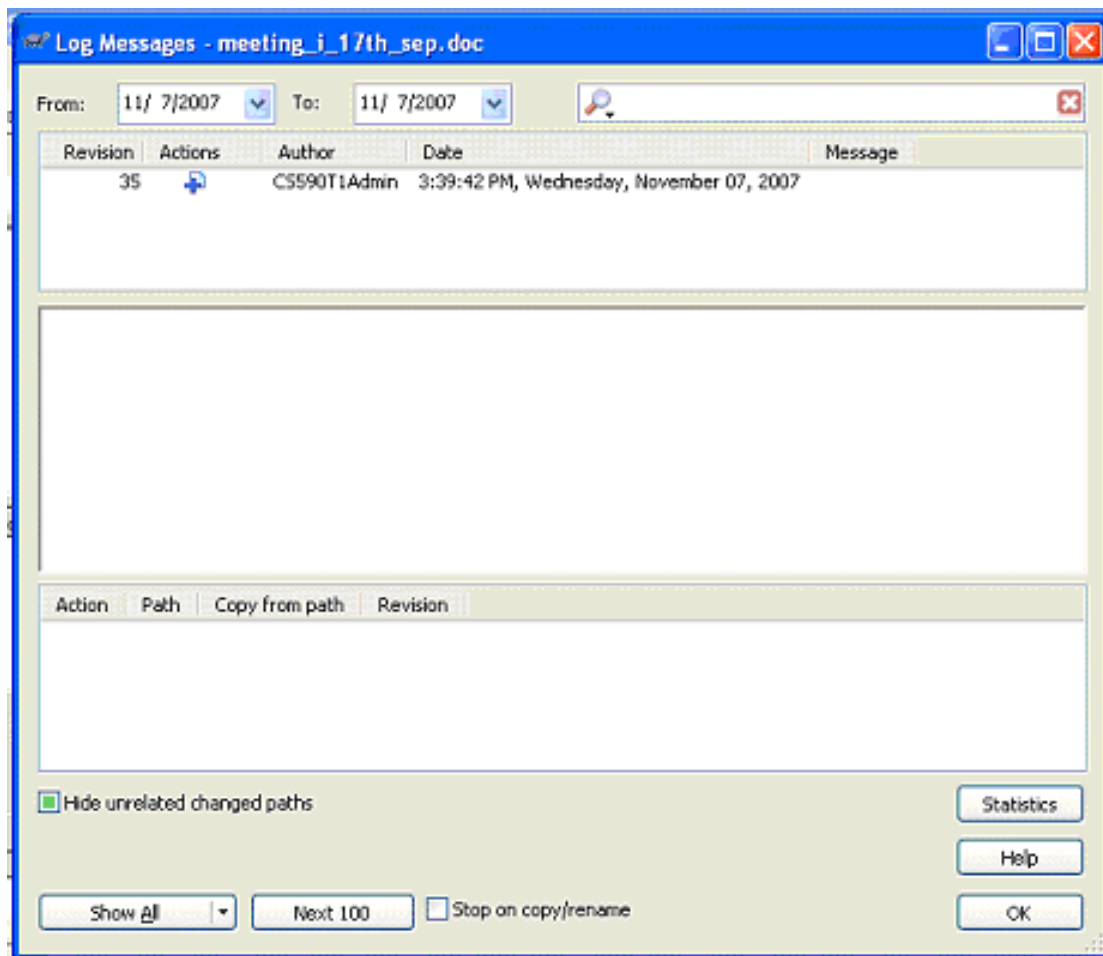
# Getting the contents of an earlier revision

If you want to view the contents of an earlier repository revision, you can do that by using the 'Update to revision' option. It gives two sub-options namely

A. Head revision – Updates the working copy with the current repository data
B. Revision – Updates the working copy with the repository data of an older version.
Note: The list of older revisions can be viewed with the help of show log button present in the dialogue box.
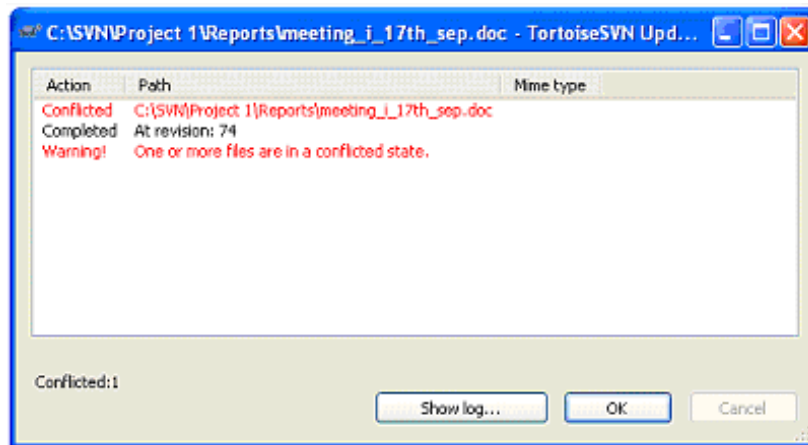
1. Right click on the file/folder
2. Select TortoiseSVN - >Update to Revision
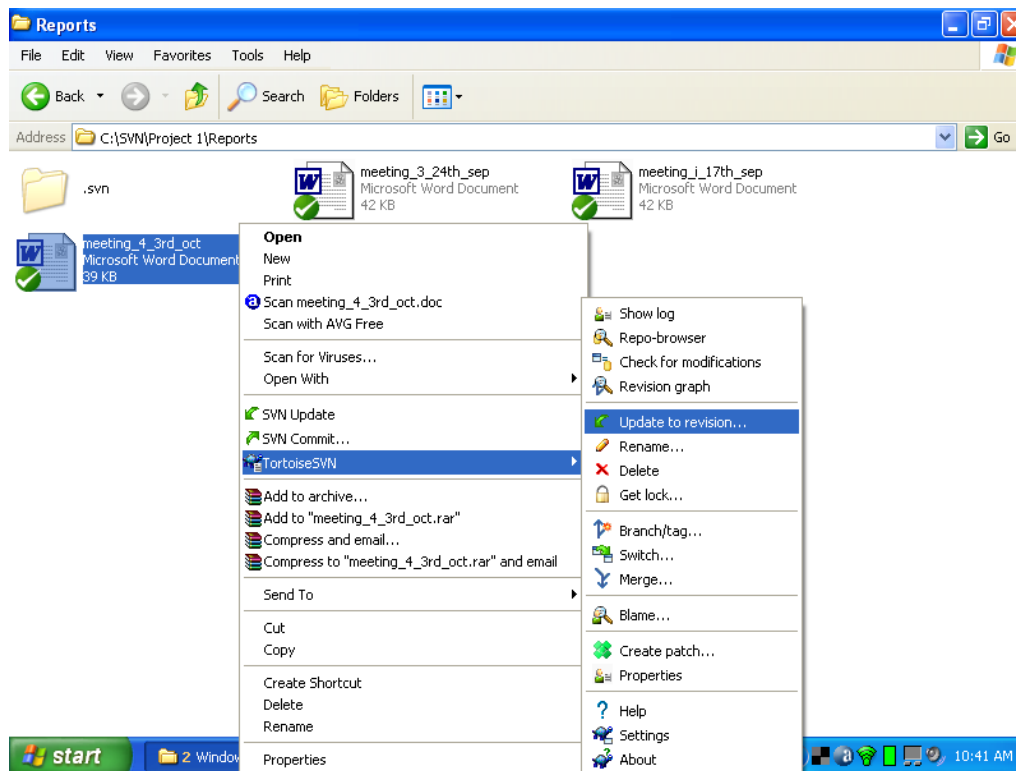


3. If you select Head revision, Click 'OK'
4. Else If you select Revision, enter the revision number viewable from the show log button into the input box, Click 'OK'

# Conflict Resolution (when same file is being modified by two or more users simultaneously)

During updates, conflict may arise that prevent TortoiseSVN from updating the repository. This type of situation occurs when two or more users have modified the same file in their corresponding working copies. After one of them has taken the SVN Commit action, when the other one tries to do a SVN Commit, the following window of conflicting state appears:



When such a conflict arise for a file, 3 additional files are created namely *filename.ext.mine*, *filename.ext.rOLDREV*and *filename.ext.rNEWREV*

1. A red exclamation overlay will appear on the file.
2. Right click on the file 'filename.ext'.
3. Select TortoiseSVN - >Edit Conflicts.
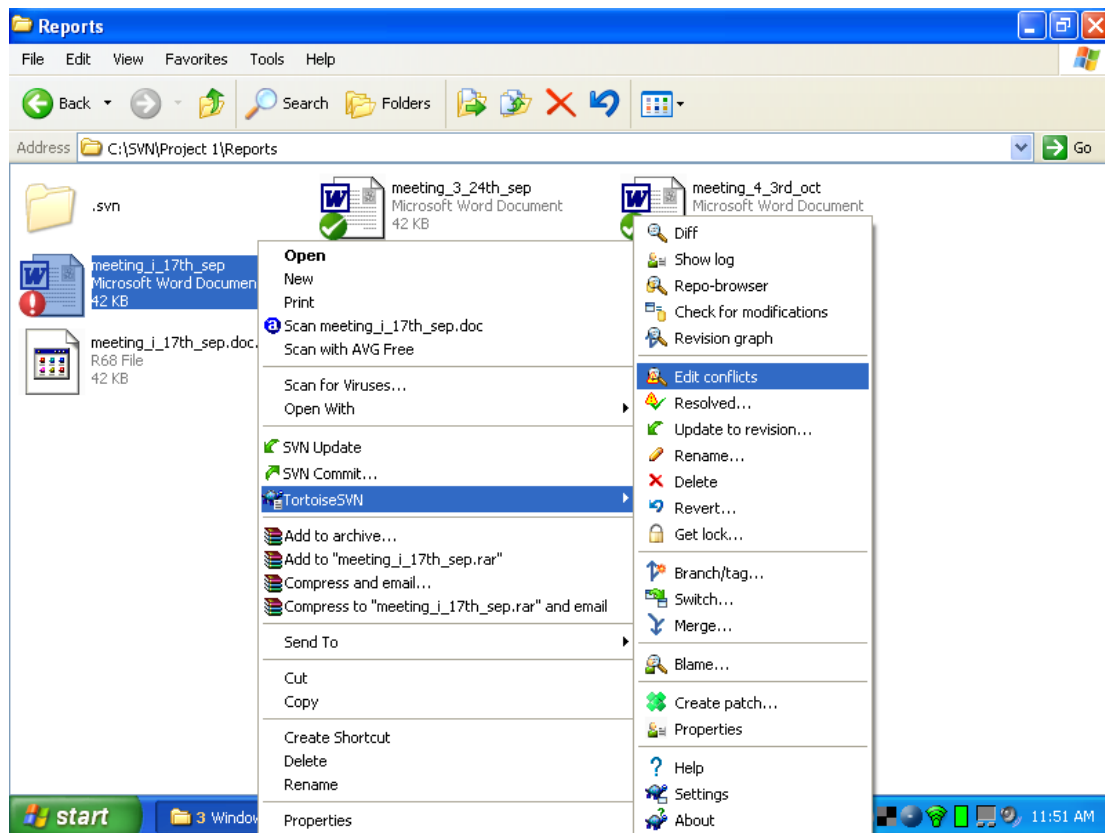
4. Resolve Conflict

The 'TortoiseMerge' window displays two conflicting versions of the files. One has changes made by you and the other is the version of the file currently residing in the repository. Right click on the changes made by you.
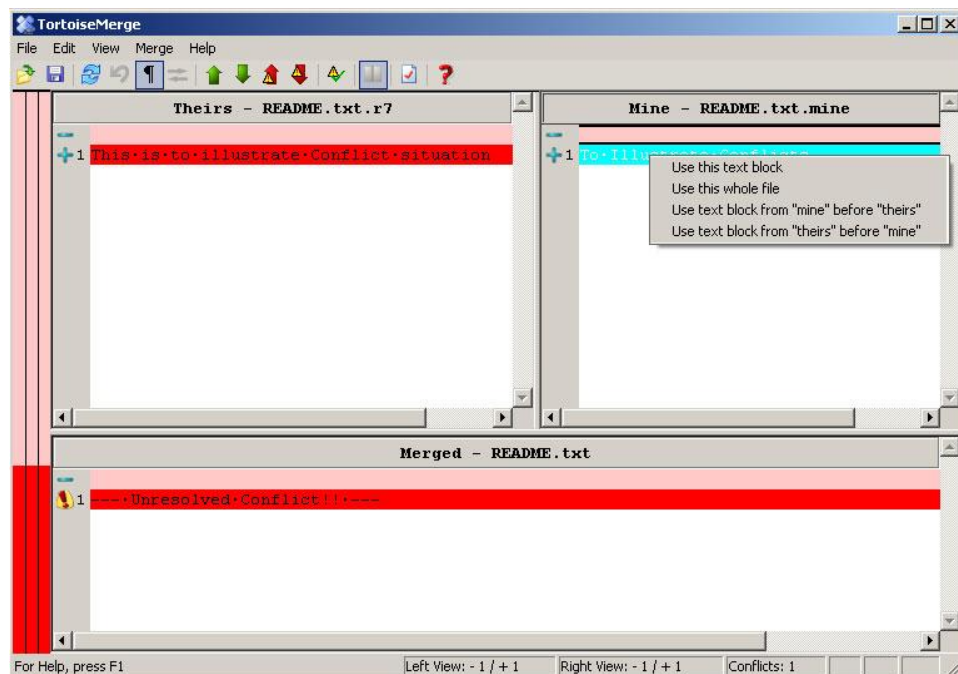
- The 'Use this text block' option will use the change done by you and will override the changes made in the previous revision.
- The 'Use this while file' option will override all the changes made in the previous revision. The changes made by you will be final.
- The 'Use text block from "mine" before "theirs"' option will append the two text snippets with your changes preceding the other set of changes
- The 'Use text block from "theirs" before "mine" option will append the two text snippets with the other changes preceding the changes done by you.

5. After choosing appropriate conflict resolving method, exit from the conflict resolve window.
6. Right click on the file 'filename.ext'.
7. Select Tortoise SVN -> Resolved. The filename.ext.r* files will get deleted.
8. Right click on the file 'filename.ext'.
9. Select TortoiseSVN - >Commit
10. Enter a log message.
11. Click 'OK'.
12. The red exclamation overlay on the conflicting file will convert to a green tick mark overlay.

## Viewing Differences between working copy and the repository

If a file is modified in working copy but SVN Commit is not done, then to view its differences with the file in the repository, we use the diff command.

1. A red exclamation overlay will appear on the modified file.
2. Right click on the file in the working copy that you want to compare with the repository's copy.
3. Select TortoiseSVN - >Diff
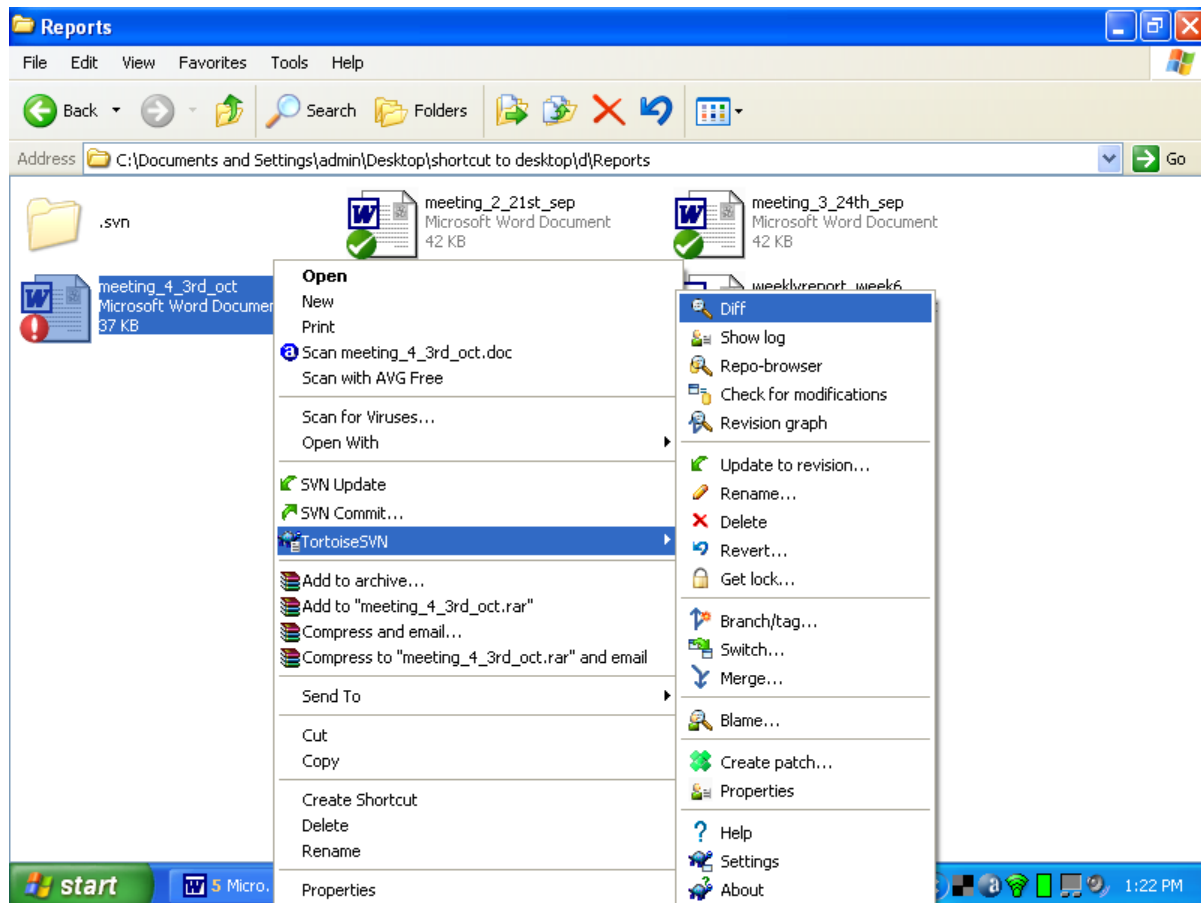


4. The file that has been modified in the working copy will open up with the differences between working copy and repositories copy mentioned.
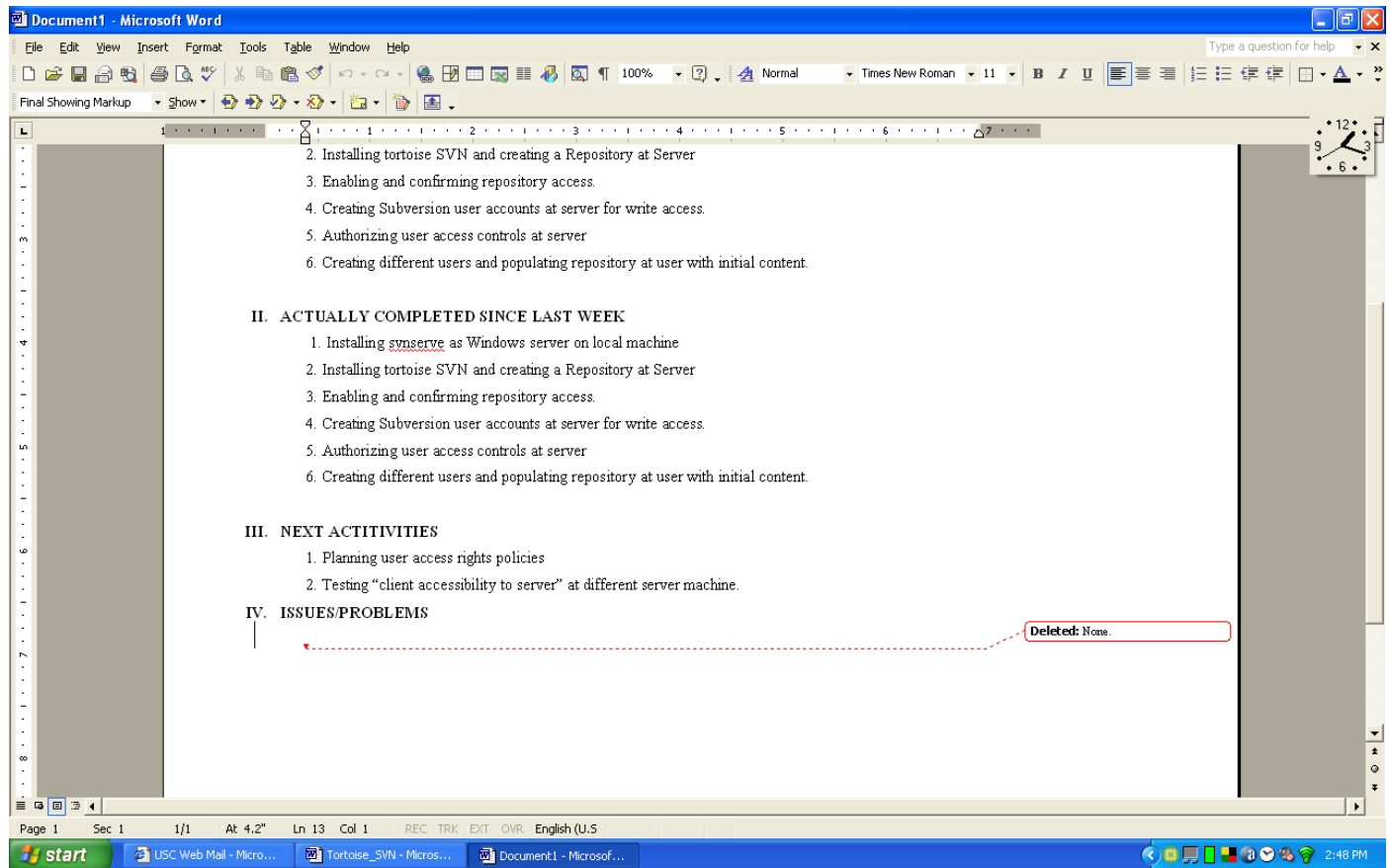
Explanation for the Screenshot: In the working copy the text "None." was deleted from the ISSUES/PROBLEMS topic in the weekly meetings file, but it existed in the repositories copy of the same file. Hence the diff command highlights the point from where the text was deleted and in a block mentions the difference i.e. "Deleted: None."

# Getting Help

To get help for using any of the Subversion commands using TortoiseSVN:

1. Right click anywhere on your Windows System
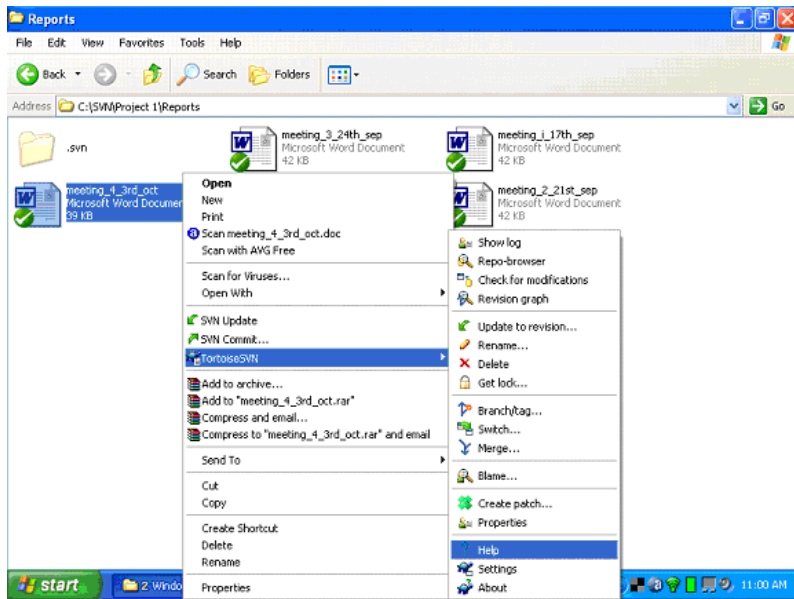2. Select TortoiseSVN -> Help



# Creating a Branch

Branching is the ability to isolate changes onto a separate line of development. This line is known as a branch. Branches are often used to try out new features without disturbing the main line of development with compiler errors and bugs. This is useful when working out a (big) change that needs to be coordinated in a team or when using a single working-copy for the work with the change.

1. Right click on the file/folder that you wish to branch.
2. Select the TortoiseSVN ->Branch/Tag option.
3. In the 'Copy (Branch/tag)' dialogue, enter the 'To URL' as 'svn://brown.usc.edu:3690/code/Fall07/Project<projectnumber>/branches/filename' where, projectnumber is the number of your project team.
4. In the 'Create copy in the repository from' section, select the revision number of the file/folder from which you want to branch

- HEAD revision in the repository: Select the option if you want to create the branch from the head revision
- Specific revision in repository: Select this option if you want to create the branch from a specific revision
- Working Copy : Select this option if you want to create the branch from the working copy revision

5. Enter a log message and click 'OK'
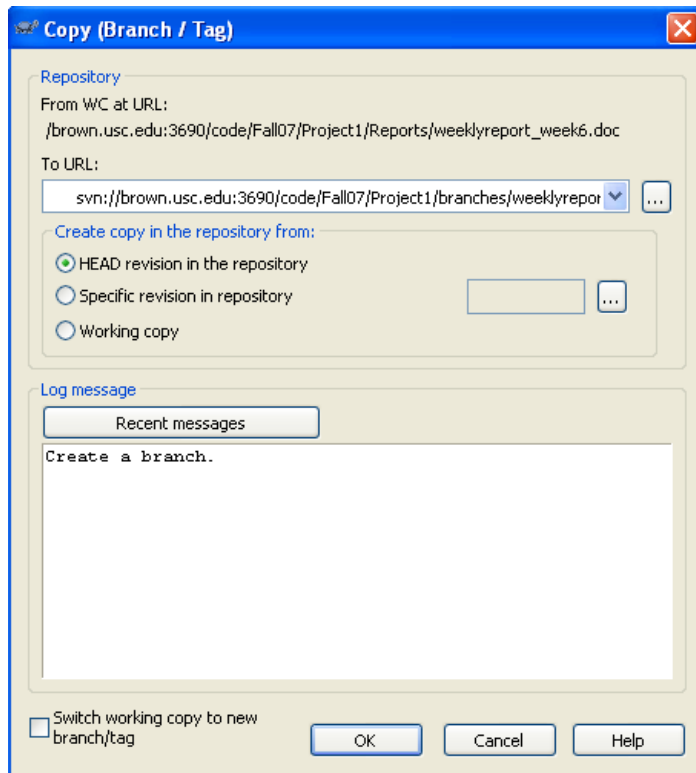


6. A success message is displayed once branch is created successfully. Click 'OK'

## Creating a Tag

Tagging is the ability to mark particular revisions (e.g. a release version), so you can at any time recreate a certain build or environment.

1. Right click on the file/folder that you wish to tag.
2. Select the TortoiseSVN - >Branch/Tag option
3. In the 'Copy (Branch/tag)' dialogue, enter the 'To URL' as 'svn://brown.usc.edu:3690/code/Fall07/Project<projectnumber>/tags/filename' where, projectnumber is the number of your project team
4. In the 'Create copy in the repository from' section, select the revision number of the file/folder from which you want to tag

- HEAD revision in the repository: Select the option if you want to create the tag from the head revision

- Specific revision in repository: Select this option if you want to create the tag from a specific revision
- Working Copy : Select this option if you want to create the tag from the working copy revision
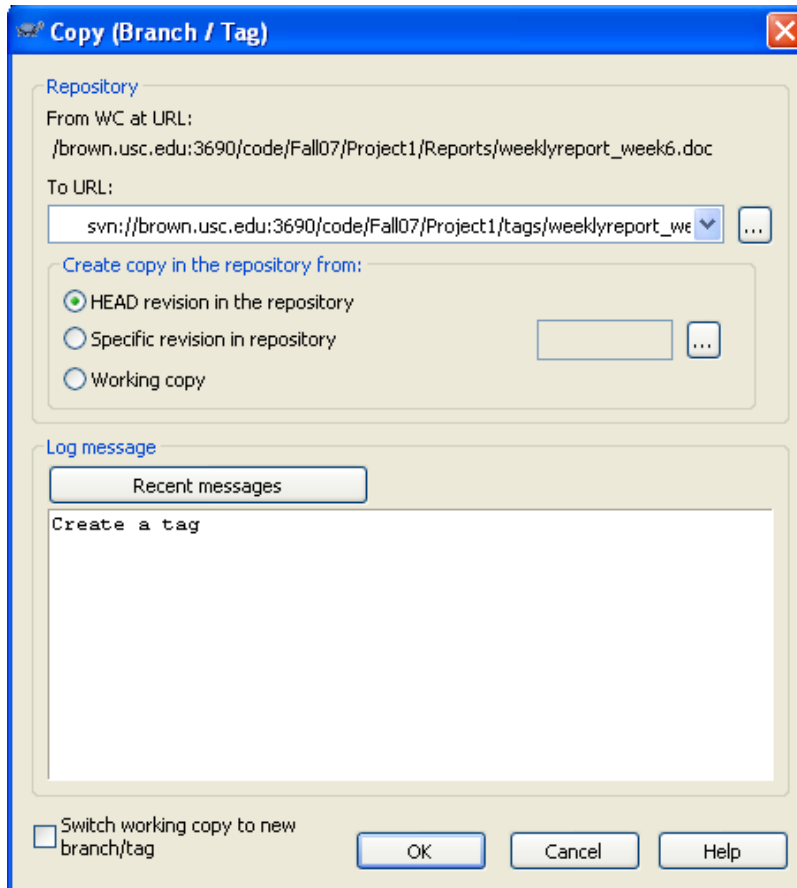
5. Enter a log message and click 'OK'



6. A success message is displayed once tag is created successfully. Click 'OK'.

## Switching between branches

The switch command transfers only the changed data of the repository to your working copy. Switch works just like Update in that it never discards your local changes. Any changes you have made to your working copy which have not yet been committed will be merged when you use the Switch option.

1.  Right click on the file/folder to which you want to apply the changes
2.  Select the TortoiseSVN - >Switch option
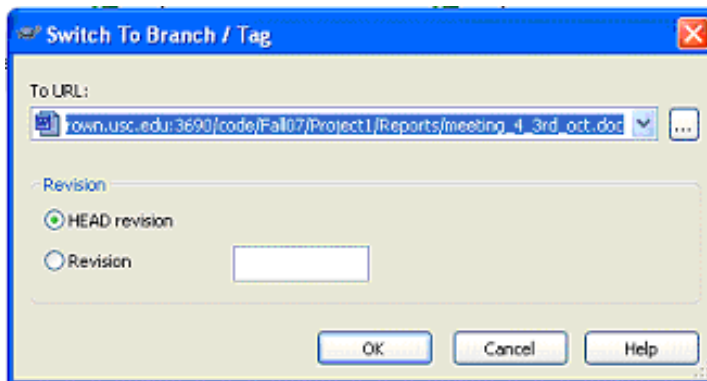


3.  Enter the URL of the file in the repository from where you want to bring in changes
4.  Choose the appropriate revision
5.  Click 'OK'

## Merge

Merge applies the differences between two sources to a working copy path.

1.  Right click on the file/folder in which you want to merge the changes
2.  Select the TortoiseSVN - >Merge option

**Merge**

Specify the revision range which you want to merge:

From: (start URL and revision of the range to merge)

svn://brown.usc.edu:3690/code/Fall07/Project1/branches

○ HEAD Revision
◉ Revision          [            ]   Show log
☐ Ignore ancestry

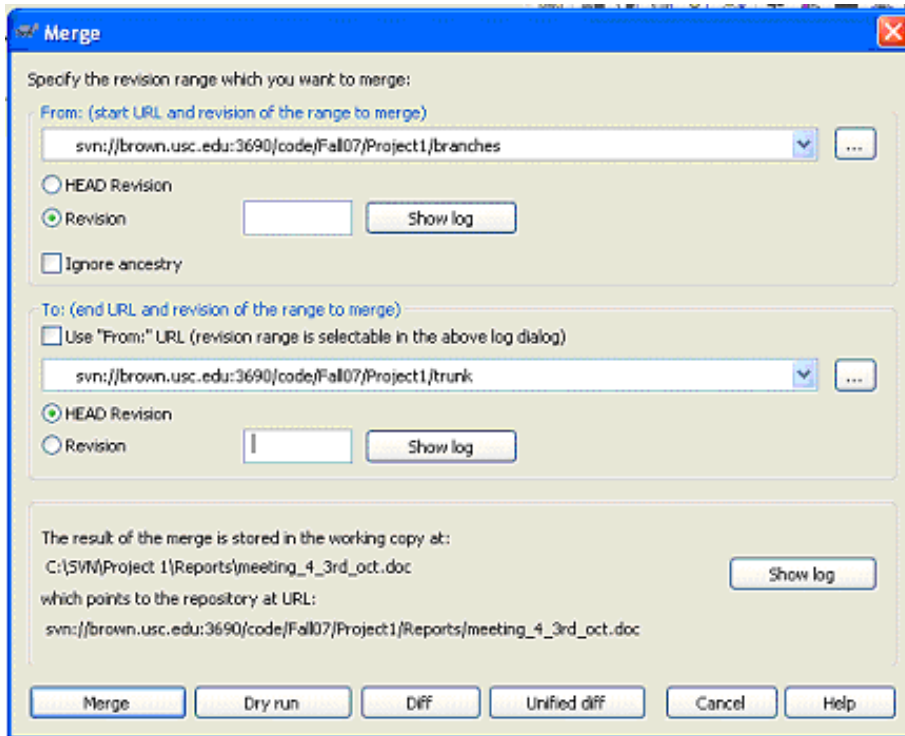To: (end URL and revision of the range to merge)
☐ Use "From:" URL (revision range is selectable in the above log dialog)

svn://brown.usc.edu:3690/code/Fall07/Project1/trunk

◉ HEAD Revision
○ Revision          [            ]   Show log

The result of the merge is stored in the working copy at:
C:\SVN\Project 1\Reports\meeting_4_3rd_oct.doc
which points to the repository at URL:
svn://brown.usc.edu:3690/code/Fall07/Project1/Reports/meeting_4_3rd_oct.doc

[ Show log ]

[ Merge ] [ Dry run ] [ Diff ] [ Unified diff ] [ Cancel ] [ Help ]

3. Enter the complete URL of the file/folder that contains the changes that you want to merge (source)
4. Enter the complete URL of the file/folder to which the changes must be merged
5. Click 'Merge' button

- As an optional step, you can choose to click the 'Dry Run' button which will perform the merge operation, but will not modify the working copy. It displays the list of the files that will be changed when you choose to 'merge', and notes those areas where conflicts will occur.
- You can also view the diff in the unified diff format by clicking 'Unified Diff' button

6. The "TortoiseSVN Merge  ... Finished!" dialogue must be displayed with message that the destination file/folder was updated with the required changes. Click 'OK'
7. Right click on the destination file/folder
8. Select the TortoiseSVN - >Commit option
9. Enter a log message
10. Click 'OK'

# Get Lock

If you get a Lock on a file, then only you can commit that file. Commits by all other users will be blocked until you release the lock. A locked file cannot be modified in any way in the repository, so it cannot be deleted or renamed either, except by the lock owner.

1. Right click on the file/folder on which you want to acquire the lock
2. Select the TortoiseSVN - >Get Lock option
3. In the 'TortoiseSVN: Lock files' dialogue, enter a reason about why you need the lock on this file/folder. Click 'OK'
   NOTE: When you wish to lock a folder, TortoiseSVN gives you the option to choose the files that exists within this folder, you can select the files that you wish to lock.
4. Once the file/folder is locked for edit by other users, the 'Lock Finished' dialogue will be displayed with a success message. Click 'OK'
5. A yellow lock overlay appears on the locked file/folder.

# Release Lock

The lock owner can release the lock from a previously locked file. Once the lock has been released, any other user can modify the file.

1. A yellow lock overlay appears on the locked file/folder.
2. Right click on the locked file/folder for which you want to release the lock
3. Select the TortoiseSVN - >Release Lock option
4. Once the file/folder is unlocked, the 'Unlock Finished' dialogue will be displayed with a success message. Click 'OK'
5. The yellow lock overlay on the locked file/folder will convert to a green tick mark overlay.

# Export

When you create a working copy of a repository some .svn directories are formed. Sometimes you may want a working copy without any of those .svn directories. For this purpose, TortoiseSVN offers the command Export to avoid making a copy and then delete all those .svn directories manually,.

1. Right click on the folder that you wish to export
2. Select the TortoiseSVN - >Export option
3. Select the location where you wish to export the folder contents in the 'Browse for Folder' dialogue. You can choose an existing location or create a new folder using the 'Make new folder' button. After selecting the location, click 'OK'
4. After the folder is exported successfully, the 'TortoiseSVN' dialogue is displayed with success message. Click 'OK'

# Relocate

To make the working copy point to a new repository, you can use the 'Relocate' option

1. Right click on the top level folder in the working copy
2. Select the TortoiseSVN - >Relocate option
3. Enter the URL to which you need to relocate to, click 'OK'
4. After the working copy is relocated successfully, the 'TortoiseSVN' dialogue is displayed with success message. Click 'OK'

# Clean Up

If your working copy gives a "working copy locked" error and becomes unusable, you must use this command to make the working copy usable again.

1. Right click on the folder on which you want to perform the 'Clean up' activity
2. Select the TortoiseSVN - >Clean Up option
3. Once the clean up is done, the 'Clean Up has Finished' dialogue will be displayed with a success message. Click 'OK'

# Repository Browser

The Repository Browser allows you to view the structure and status of the repository.

1. Right click on the file/folder
2. Select the TortoiseSVN - >Repo-browser
3. The repository browser must open which displays the directory structure of the working copy
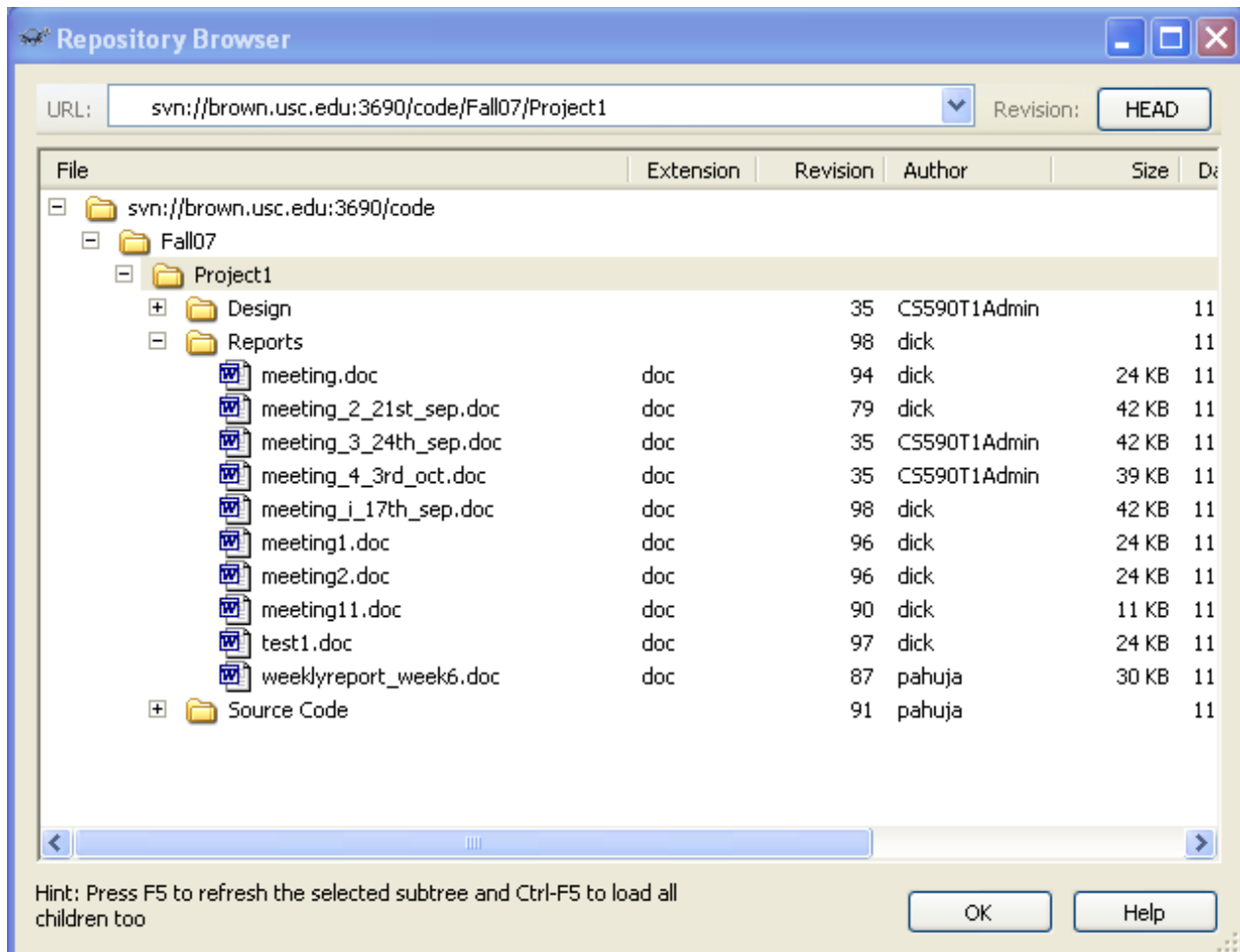
# Check for modifications

It is similar to the diff command and tells you about the modifications that have taken place in the file/folder in the working copy compared to the file in the repository.

1.  Right click on the file/folder
2.  Select the TortoiseSVN - >Check for Modifications
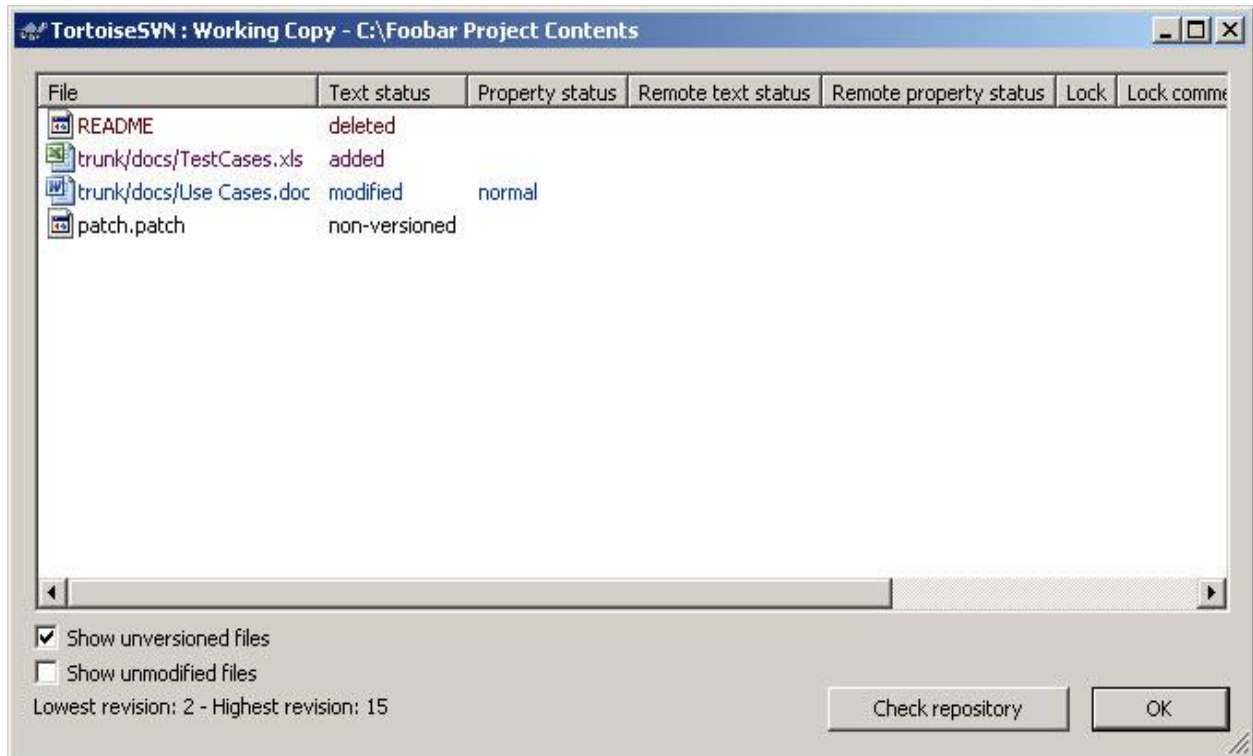


3.  When using this option for a folder, the window that opens displays all the files that were modified since the last commit. In case of a file, it just gives the information about the user who made the changes

# Settings

To set your Subversion specific settings, use the 'Settings' option

1.  Right click any where on your system
2.  Select the TortoiseSVN - >Settings option
3.  You can decide on the look and feel, network, external programs, and saved data from here. After setting the choosing the required options, click 'OK'.

# Revision Graph

This command analyses the revision history and attempts to create a tree showing the points at which copies were taken, and when branches/tags were deleted.

1. Right click on the file/folder
2. Select the TortoiseSVN - >Revision Graph option
3. The revision graph window displays the number of versions the file/folder has been through
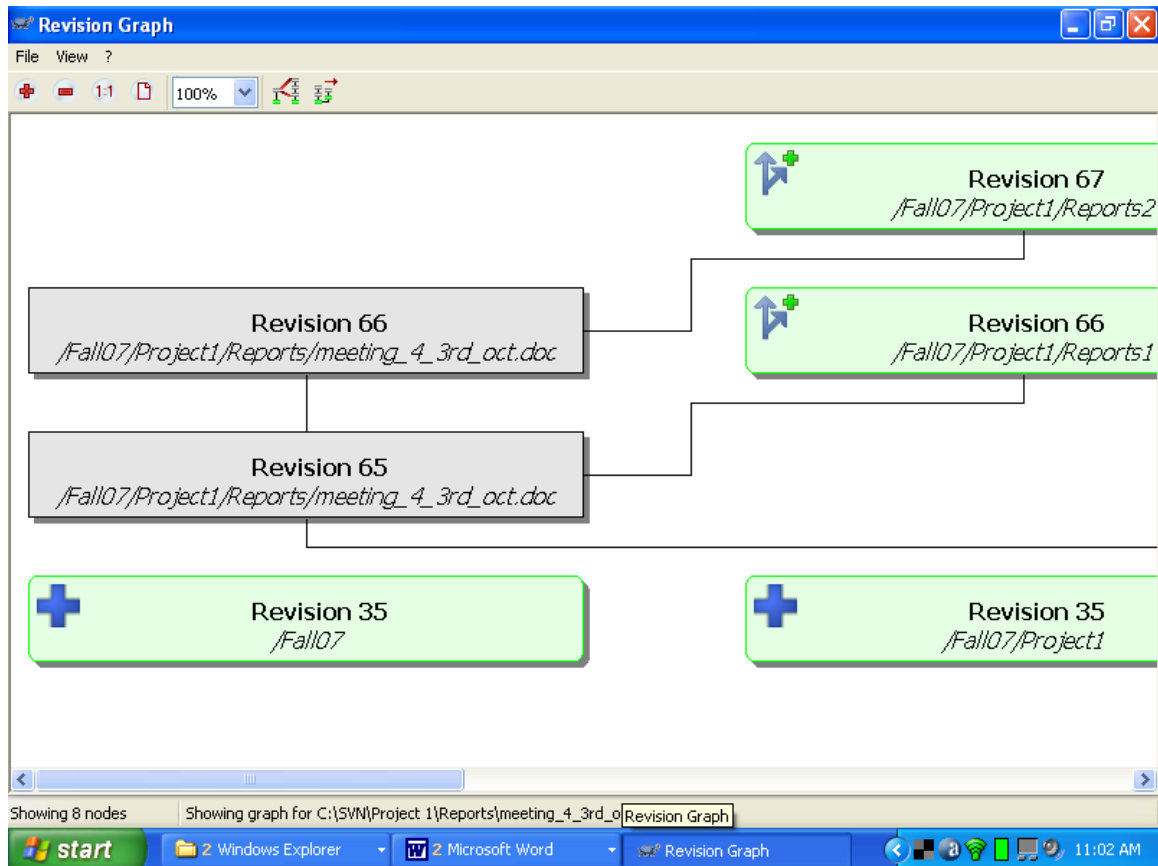


# About

1. Right click any where on your system
2. Select the TortoiseSVN - >About option
3. Information on the installed version of TortoiseSVN must be displayed. Click 'OK'

# Important points to remember:

- You may find that not all of these icons are used on your system. This is because the number of overlays allowed by Windows is limited to 15. Windows uses 4 of those, and the remaining 11 can be used by other applications.

- Normal, Modified and Conflicted are always loaded and visible.

- Deleted is loaded if possible, but falls back to Modified if there are not enough slots.

- Read only is loaded if possible, but falls back to Normal if there are not enough slots.

- Locked is only loaded if there are fewer than 13 overlays already loaded. It falls back to Normal If there are not enough slots.

- Added is only loaded if there are fewer than 14 overlays already loaded. It falls back to Modified if there are not enough slots.