How to Setup Subversion Source Control Repository

We need to use **Tortoise SVN client**, which is a windows based client that integrates with your windows explorer.

**What is source control**? Source control is a code storage, code versioned and it facilitates multiple developers

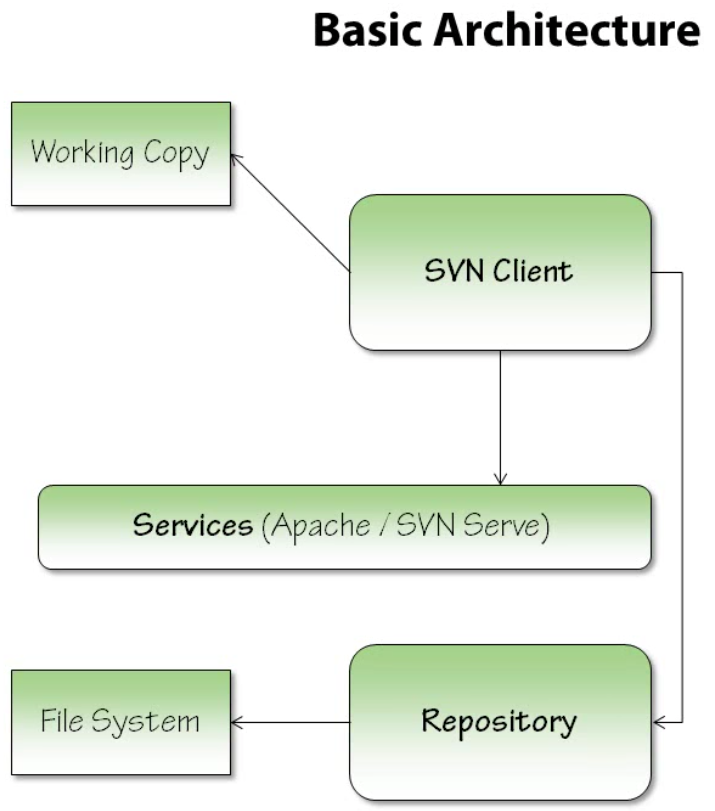
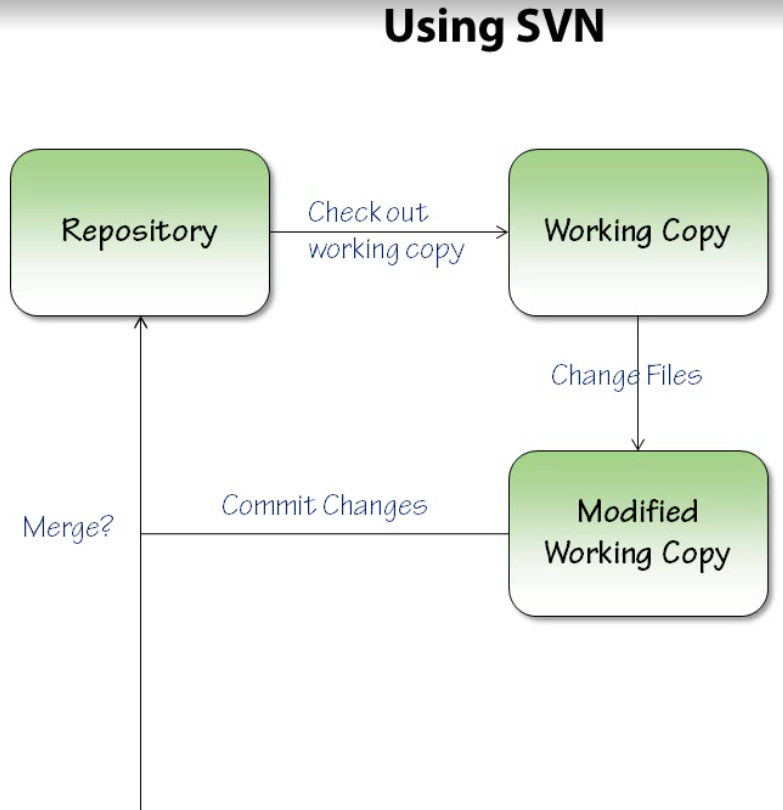
**Locking** prevents 2 or more people making changes on a same file. You have to wait for someone to check-in their own file before we can make the change. Only one person can change code at a time.

**SVN** is a merging source control system where we don’t lock files when editing but instead we download a local working copy of the code and make our changes. Once changes are made and finalized, we then submit a pull request and followed by a manual merge.

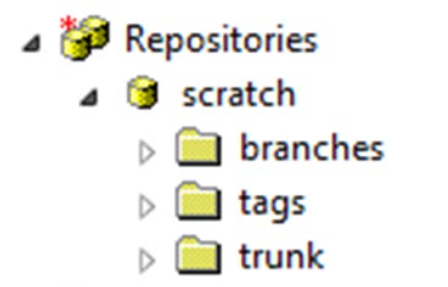
**GIT** is a distributed source control system where instead of one central source where the code lives, developers has its own copy of the repository where they make changes. Those changes/change sets will then be pushed to different repository.

**Commits are Atomic** – This means that if you make a changes on several files on your source and then you commit those or check them in (this is the way you tell the server that you’re done & you want your change to go to the server). It would be all or nothing operation which means either all of those updated files are check in and committed or none of them are.

**Branching and Tagging Code Copy** – Splitting code into a new branch like and put a label and tag on it.

**Repository Structure**



**Branch** – It a code that you maybe create a new code line or release or new version. It is a way to segregate your code and make the changes locally. It uses a copy a code and only the difference. Branches are like new trees and has the same rules applied.

**Tags** – A feature in subversion which act as a way to name a revision. It looks like a branch and acts like a branch but its read only. It doesn’t take additional space since its only copy the differences. It’s just a level or a name (ex: release2.5)

**Trunk** – The main code line where active development is happening. You are getting and checking files here. You may branch into a release or a feature but everything comes back to trunk (main project)

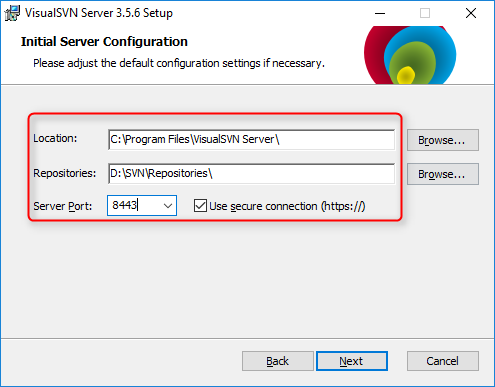
Setting up SVN Server

1. Go to <http://subversion.apache.org> or <http://subversion.apache.org/packages.html>
2. Download Visual SVN at <http://subversion.apache.org/packages.html#windows>

**Install VisualSVN Server Enterprise Edition**

Perform the following steps to freshly install VisualSVN Server Enterprise Edition:

1. Start the VisualSVN Server installation package and accept the terms of the License Agreement.
2. Choose to install **VisualSVN Server and Management Console**.
3. Click the **Enterprise Edition** button.
4. Enter your **evaluation license key** for VisualSVN Server Enterprise Edition.
5. Consider initial configuration settings and adjust them if needed.
6. Click the **Install** button to install VisualSVN Server Enterprise Edition.



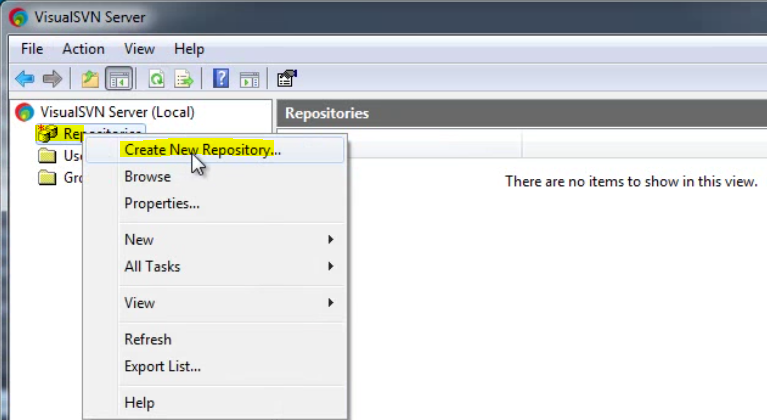
**Upgrade existing VisualSVN Server instance to the Enterprise Edition**

Perform the following steps to upgrade existing VisualSVN Server instance to the Enterprise Edition:

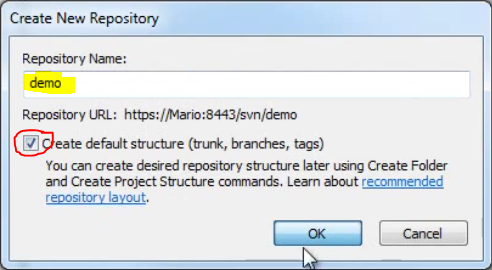
1. Start **VisualSVN Server Manager**.
2. Choose **Action | All Tasks | Update License** main menu command.
3. Enter your **evaluation license key**.
4. Your instance of VisualSVN Server will be upgraded to the Enterprise Edition.

Creating Repository

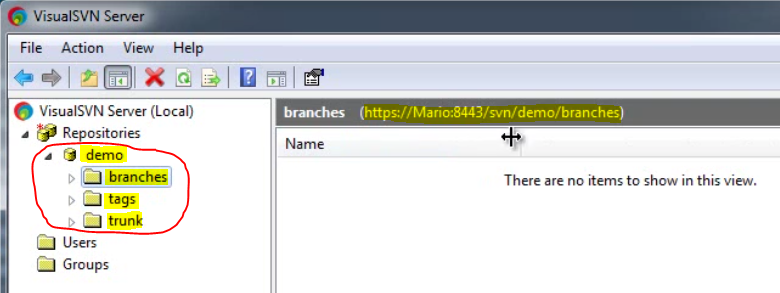
1. Open Visual SVN Server
2. Right Click “Repositories” 🡪 Create New Repository



1. Give a new repository a name



1. Select “Create Default Structure”.
2. Click OK. You will now see the new repository created.



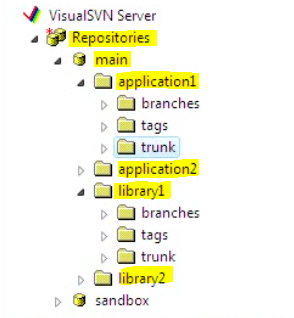
Repository Security and Hooks

1. Open VisualSVN Server and Expand Repositories
2. Right Click the target repository 🡪 Select “All Tasks”
3. Select “Manage Hooks”

What is Hooks? Before you start to commit, pre-commit and post-commit your code you can create or insert test (ex: Unit Test) or whatever you want to do (ex: Check for comments) before that code can be committed. Commits will then fails if some hooks implemented is not meet or satisfy.

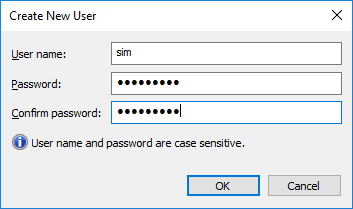
What is Locks? This feature is locking a file for editing. User needs to check it in first before it can be edited again. This is good for binaries

Suggested Repository Layout



Create Repository User

1. Open VisualSVN Server and Expand Repositories
2. Navigate to “Users” folder.
3. Right Click the folder 🡪 Create New User

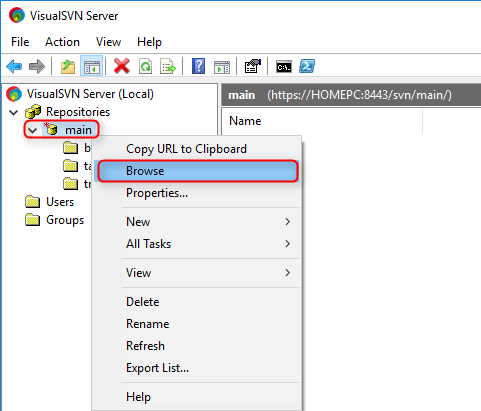


Create Folder and Project Structure

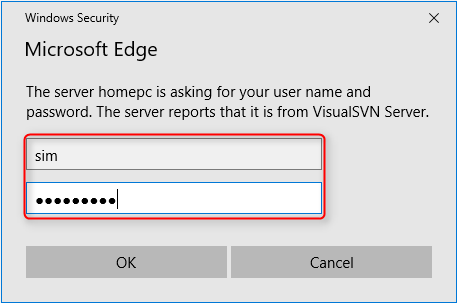
1. Open VisualSVN Server and Expand Repositories
2. Right Click the target repository 🡪 Select “New”
3. Select either “Folder” or “Project Structure” to create

Browse or Open Repository

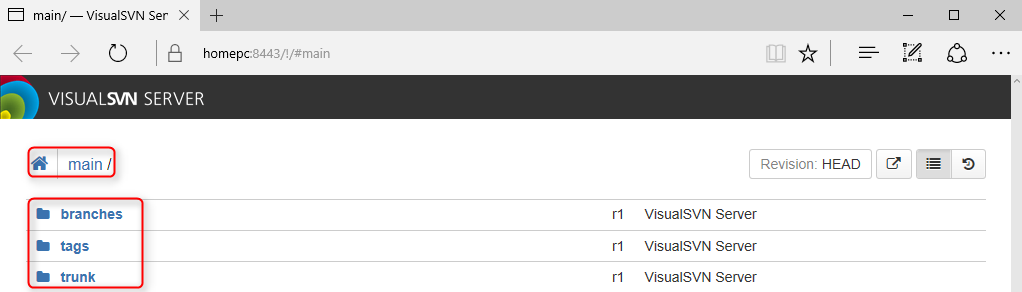
1. Open VisualSVN Server and Expand Repositories
2. Right Click the target repository 🡪 Select “Browse”



1. Enter the SVN user name and password you have created https://HOMEPC:8443/svn/main/

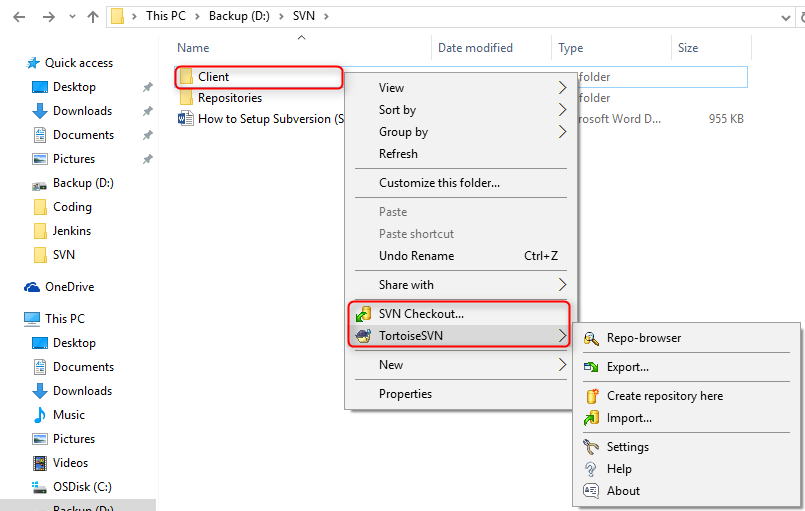


1. Click OK, you will now be able to login and then browse the folders in the repository.



Setting up SVN Client

We already setup our SVN Server and SVN Repository and now we need to install an SVN client.

1. Go to <https://tortoisesvn.net/downloads>
2. Download and install TortoiseSVN client.
3. Verify the installation by opening windows explorer and right click any folder.
4. You can now see options for “SVN Checkout” and more options under “TortoiseSVN”.

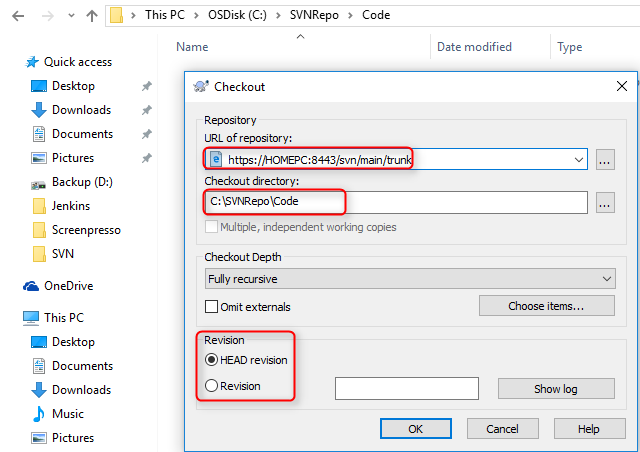
Checkout Code

To checkout a working copy of the source code from repository, you need to do the following:

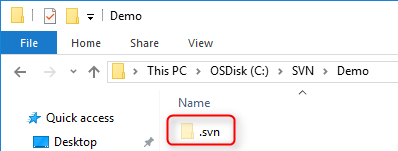
1. Create a local folder in your computer ex: C:\SVNRepo
2. Right Client from inside “C:\SVNRepo” and select “SVN Checkout”.
3. Enter the repository URL (ex: <https://servername:8443/svn/main/trunk>) where “main” is the name of the repository.

**Important**: You don’t need to check out the whole repository. You just need to get a copy of the trunk folder.

1. Enter checkout directory (ex: C:\SVN\LocalCopy)
2. Under revision, select “Head Revision” 🡪 Head means the current latest version.
3. Click OK.



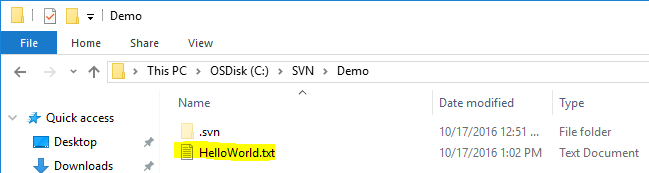
1. You will now see Trunk copied over to your local checkout directory ex: C:\SVN\Demo. If you see “.svn” hidden folder that means that trunk is cloned. It’s just it has nothing inside and it’s empty.



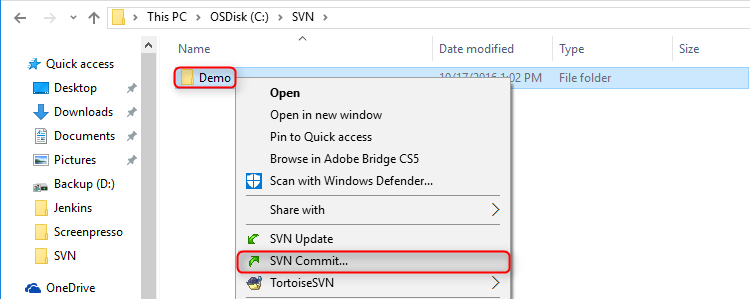
Committing Code

After you make a code change, you are now ready to make a check-in or commit those changes back to the trunk or master. To commit a new copy of the source code from your local repository, please do the following:

1. Make code changes and then save it (Ex: We just added a new file “HelloWorld.txt”)

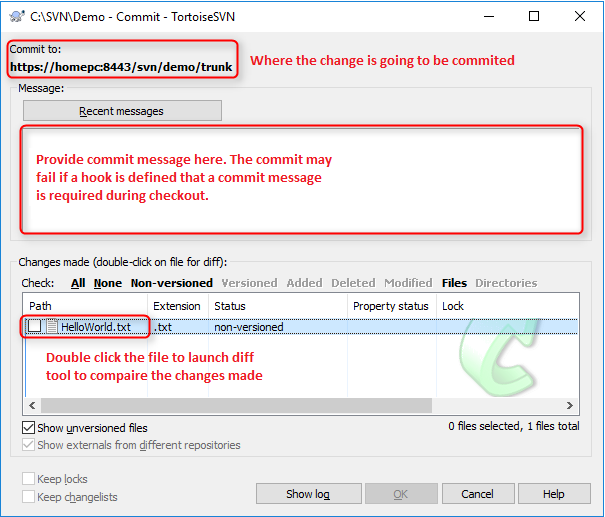


1. On the local folder where you saved a copy of the edited file, navigate to the top level folder and then do Right+Click the folder and select “SVN Commit”.



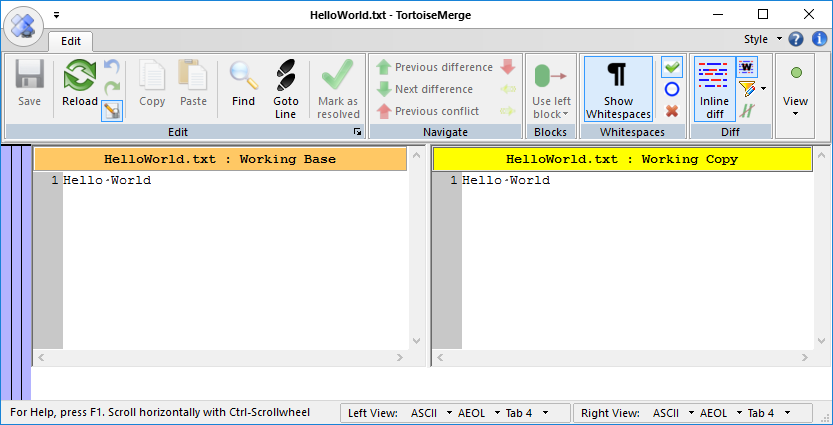
**Important**: We typically commit at a higher level directory to make sure that you don’t missed anything.

1. The commit window will open.

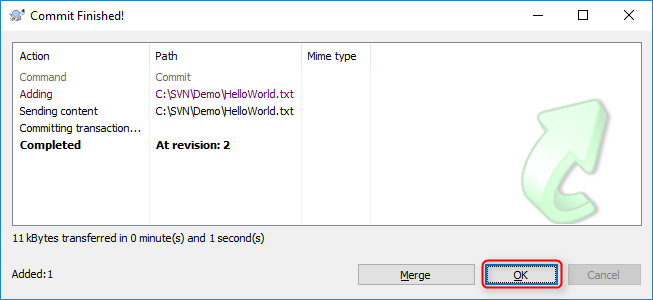


**Important**: Please attention to what folder you are committing to. We don’t want you to avoid making a mistake committing to a wrong folder or repository.

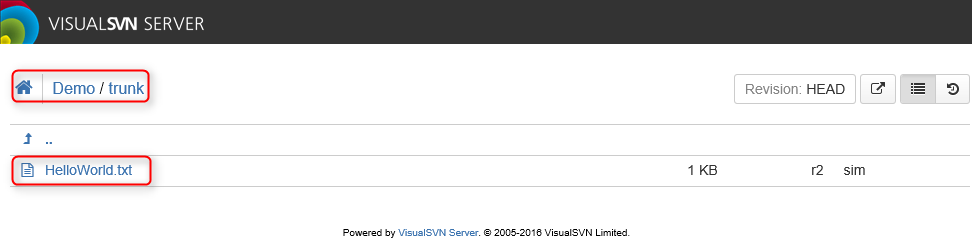
1. Double click the file to launch diff tool for you to compare the changes made.



1. Click OK button on the commit window, after providing the commit message.

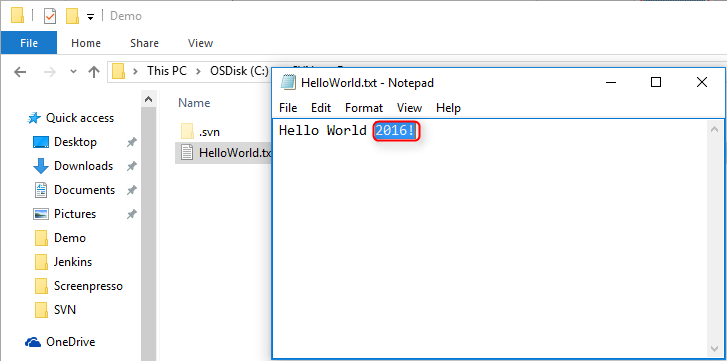


1. Verify if “HelloWorld.txt” is merged into the trunk folder by going to the SVN repository URL ex: <https://HOMEPC:8443/svn/Demo/trunk> and provide your SVN username and password.

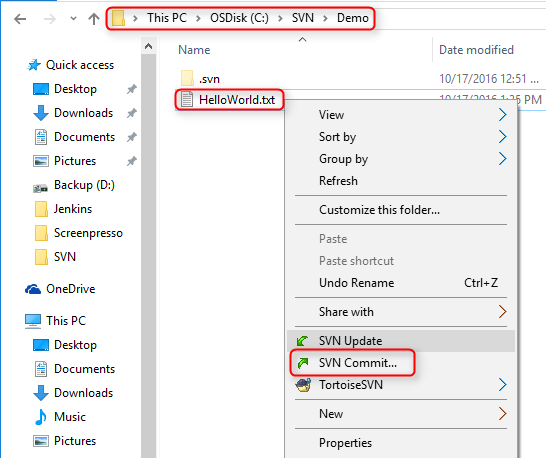


**Individual File Commit**

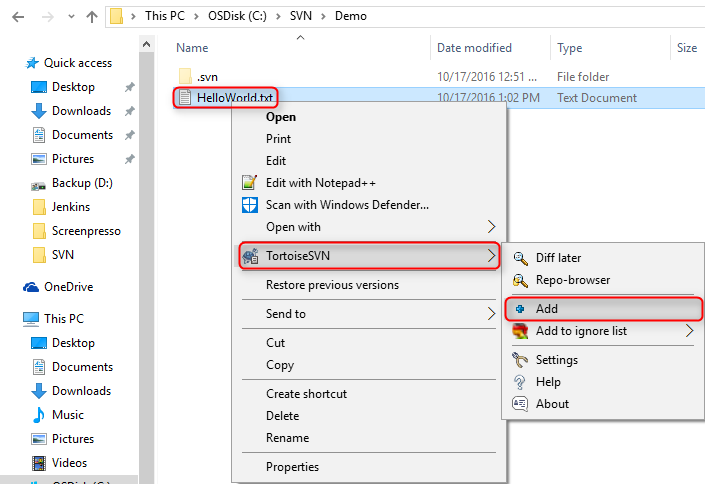
1. Make code changes on the file and then save it.

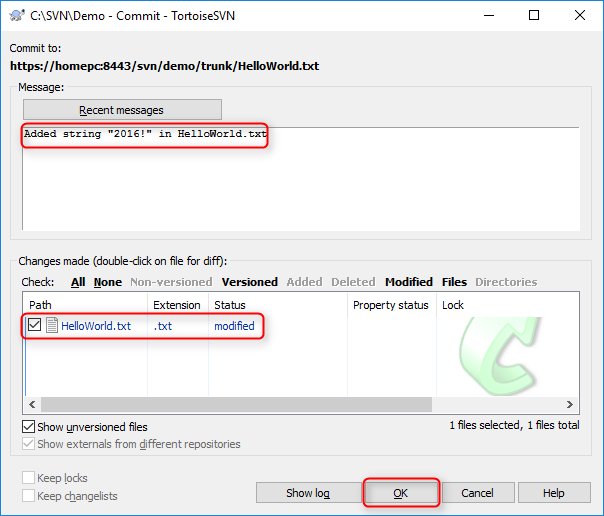


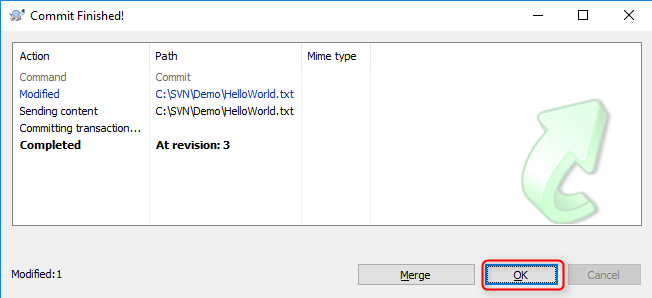
1. Right Click the file and select “TortoiseSVN” 🡪 “Add” if it’s a brand new file. Select “SVN Commit” if changes are made on the existing file.



1. Provide commit message comment and click OK.
2. Right Click the file and select “TortoiseSVN” 🡪 “Add” if it’s a brand new file.

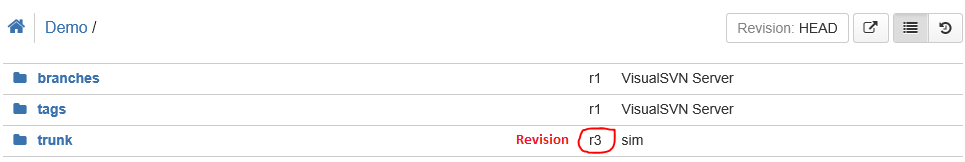




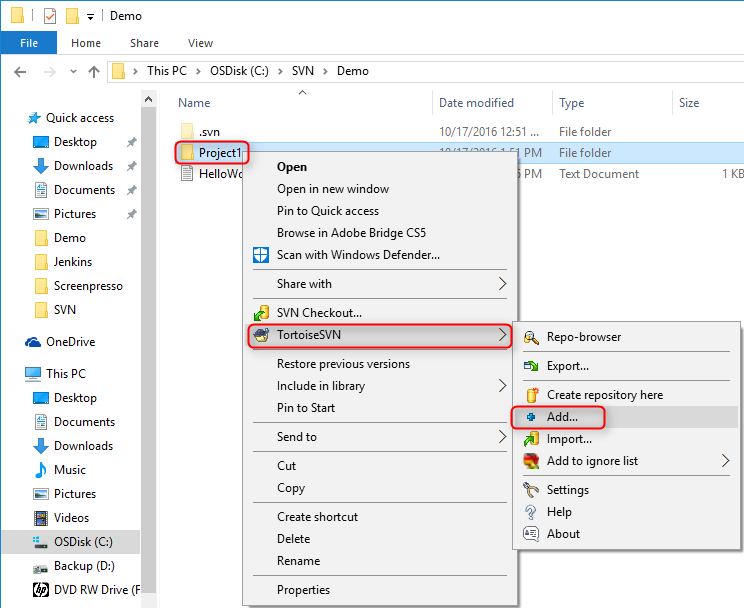


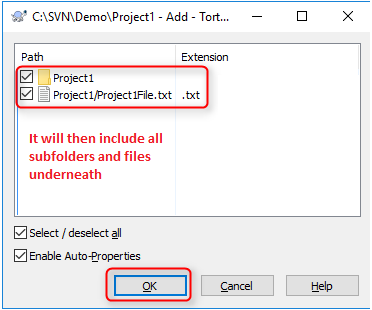
Note: You tag the file (+Add) it will then add the “HelloWorld.txt” file into the repository when I commit.

1. Go to the main repository URL ex: <https://HOMEPC:8443/svn/Demo/> and check the current trunk revision number.

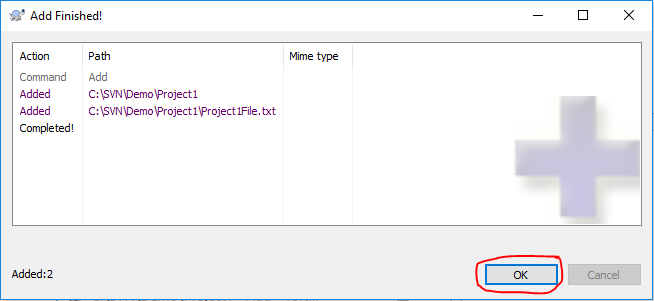


**Adding a Folder and File to Commit**

1. Create the folder and add a file.
2. Right Click the file and select “TortoiseSVN” 🡪 “Add”.
3. Select the folders and files to be committed and click OK.

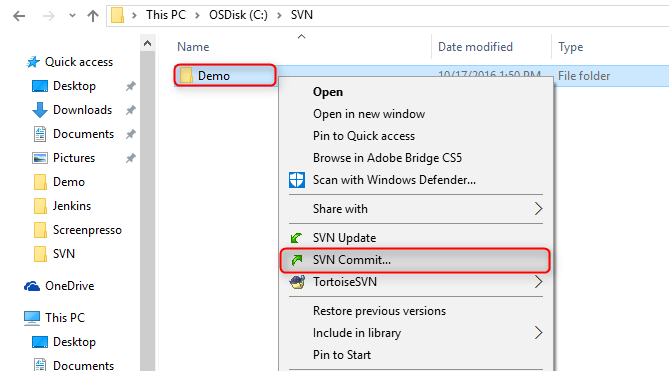


1. Commit notification window will show. Click OK

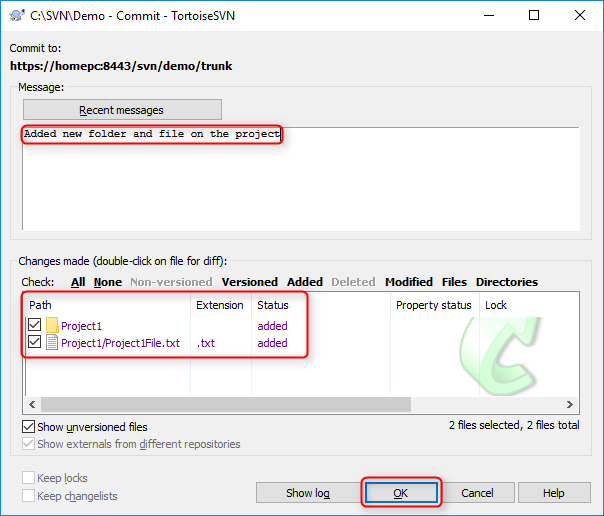


**Note**: All these changes are only happening locally, until we do a commit on the top level folder.

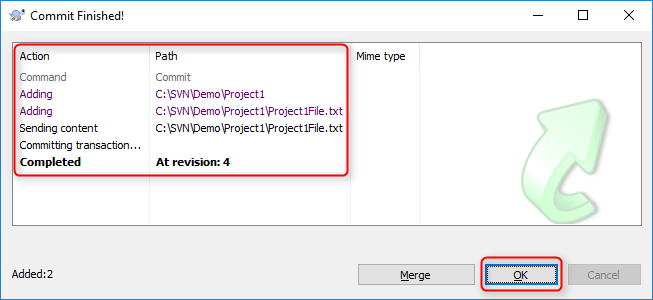
1. Navigate to the local top level folder of your newly updated code or folder. Right Click the Folder and select “SVN Commit” to commit the change.



1. Provide commit message.



1. Commit notification window will show. Click OK

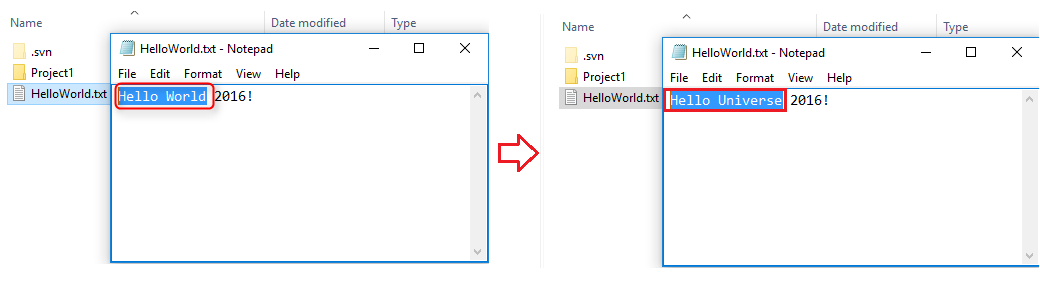


1. Check the SVN repository URL ex: <https://HOMEPC:8443/svn/Demo/trunk> and provide your SVN username and password to check if the files and folders are pushed.

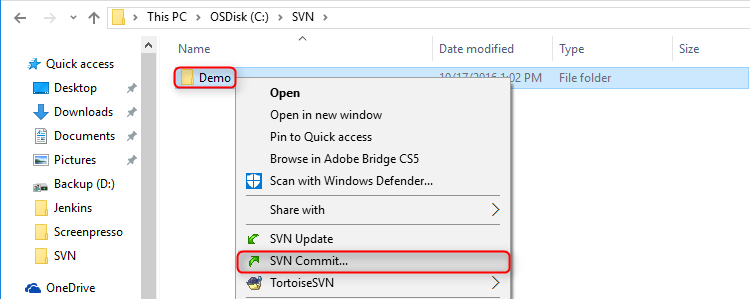


Updating a File

1. Make code changes on the file and then save it.

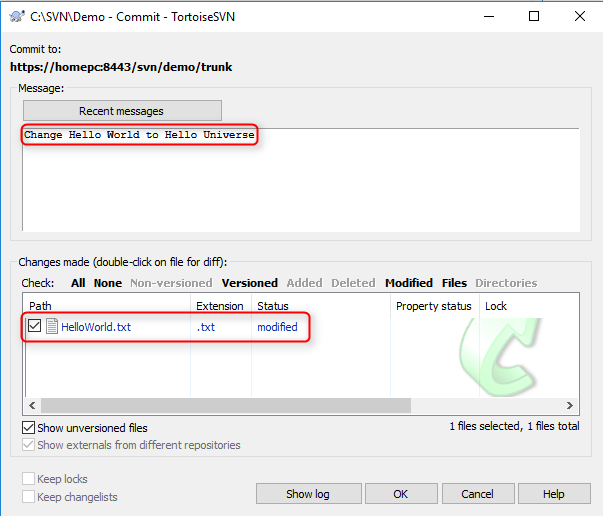


1. Navigate to the top level folder and then do Right+Click the folder and select “SVN Commit”.

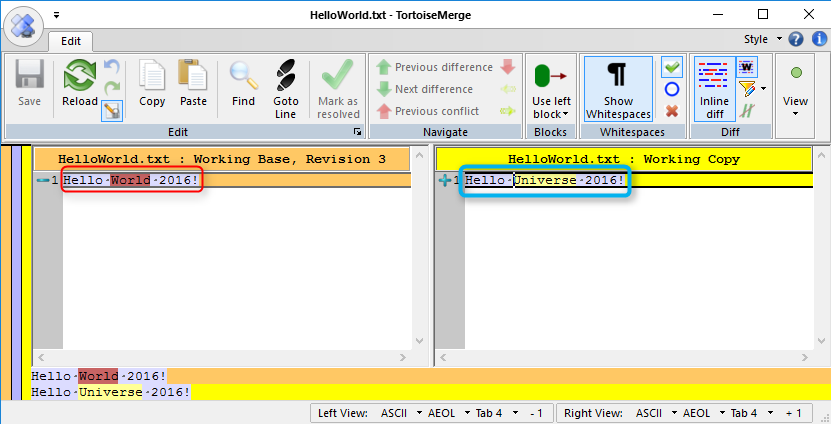


**Important**: We typically commit at a higher level directory to make sure that you don’t missed anything.

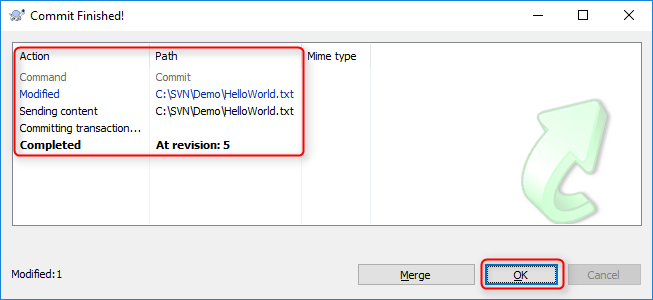
1. Add your commit comments to specify the change you’ve made.



1. Double Click the File to launch diff tool for you to compare the changes made.



1. Click OK and revision notification window will appear.



1. Check the SVN repository URL ex: <https://HOMEPC:8443/svn/Demo/trunk> and provide your SVN username and password to check if the files and folders are pushed.



Basic Commands

Branching and Merging

Best Practices