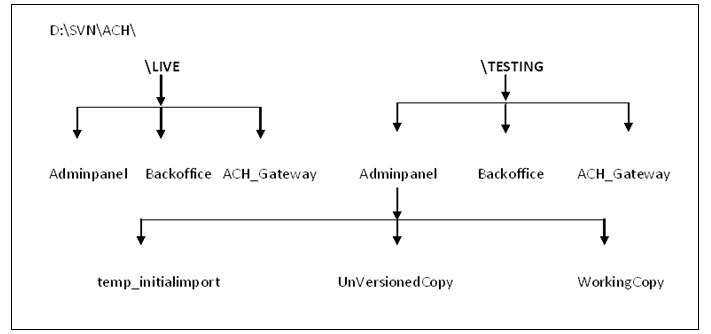
SVN

**Steps to follow:-**

1. Create a following folder structure on local machine as per the structure we maintained on subversion (Ex: D:\SVN\ACH). Now we are using 10.70 as subversion server. Later it will be changed to 111.22 (Code Live)



1. At very first, download the source code from Web server to Unversionedcopy folder of local.
2. We have to create a directory structure within the temp\_initialimport like below

/trunk

/branches

/tags

1. Copy the source code of Unversionedcopy to trunk folder of temp\_initialimport.
2. Right click from temp\_initialimport and import the new projects to Trunk of SVN.

Ex: https://10.70.1.12:8444/svn/ACH/TESTING/ACH\_Gateway/trunk

1. Once imported, delete the folders (temp\_initialimport, Unversionedcopy) from client machine.
2. Checkout the source code from trunk to working copy.
3. Never use trunk of SVN folder to development because trunk is a Main line of the development which holds current live source code.
4. Create a SVN tags from trunk with versioning (Ex: <https://10.70.1.12:8444/svn/ACH/TESTING/ACH_Gateway/tags/version1.0.00>) by using Branch/Tag option
5. Tick the option “switch working copy to new branch/tag” .So that we can’t commit anything on trunk.
6. Create a branch which means separate line of development from tags Ex: https://10.70.1.12:8444/svn/ACH/TESTING/BackOffice/branches/1.0
7. We should commit the changes to branch after testing completed on web server.
8. Once branch is ready, we should do merge with trunk and create new tags from trunk.
9. Export the Working copy into web server. Before export, download the web server code and overwrite with working copy. Both should be same.

**NOTE:**

1. Before move into web server, test everything on client machine and then export.
2. No need to create repository in client machine
3. To reduce confliction, every time we want to work, make sure **SVN Update first**
4. Alter the Global ignore pattern of the Tortoise setting with the following

\*.o \*.lo \*.la \*.al .libs \*.so \*.so.[0-9]\* \*.a \*.pyc \*.pyo \*.rej \*~ #\*# .#\* .\*.swp .DS\_Store \*bin \*obj \*suo \*.user \*.tmp \*.TMP

1. Don’t create any folder inside the trunk of client. We can put the code directly into trunk
2. Don’t touch the trunk for any new development.
3. Branch should be taken from latest tags version only.
4. We should take tags after every release
5. Instead of checkout just do switch your working copy to branch where you are working for.
6. If we merge Branch A to Branch B means, the source should be in Branch B. Please make sure this.

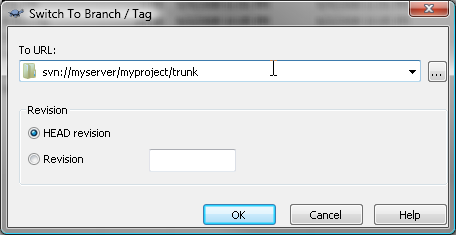
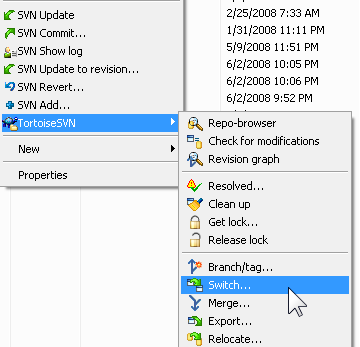
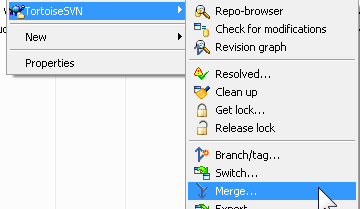
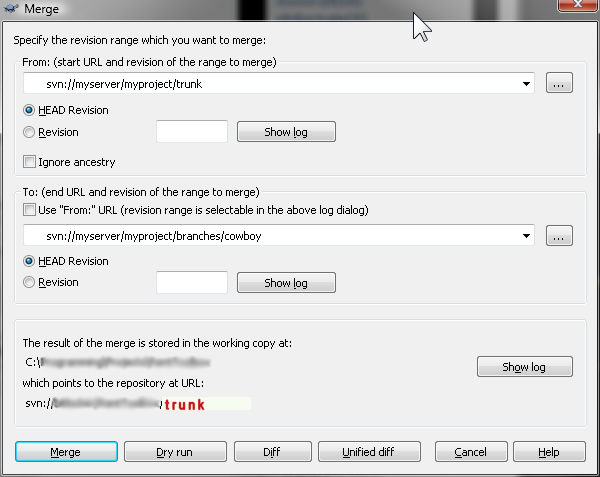
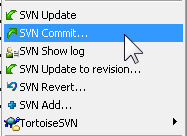
**TRUNK** 🡪 Main line of the development which holds current live source code

**TAG** 🡪 Released copy

**BRANCH** 🡪 Copy of code derived from a certain point in the tags that is used for development

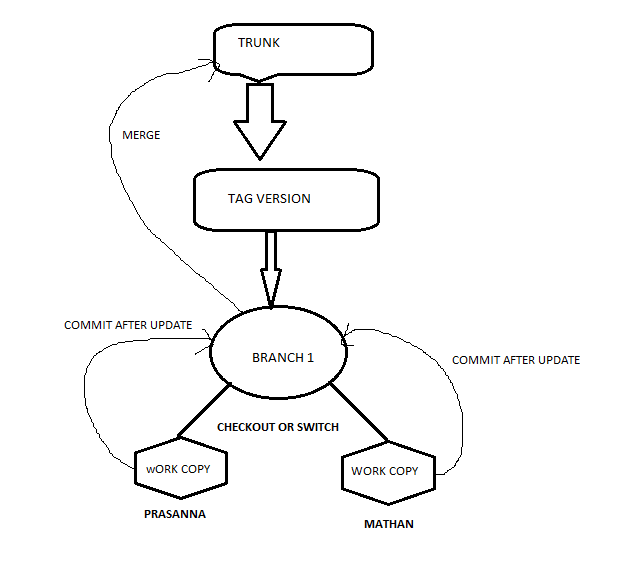
**Merge Sample:-**

**Merge branch to trunk (branch->trunk)**

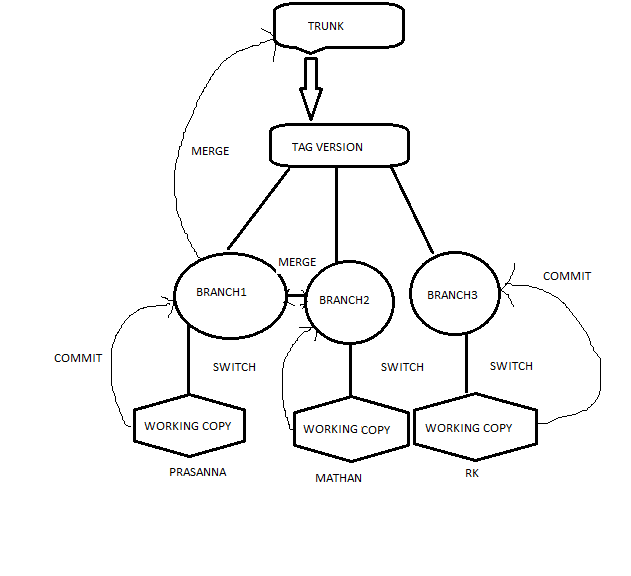
1. Commit all code in your working directory.
2. Switch your working directory to the trunk:   
     
   
3. Next, Merge...   
   
4. In the merge window, we need to make sure we are going "FROM" the trunk "TO" our branch.  The reasoning behind this is that we are doing a DIFF between the trunk and the branch.  Or, starting "**FROM**" the trunk's current state "**TO**" our branch's state.  The trunk is the starting point for the branch changes.   
     
   
5. Now that the diff has been calculated, the results are stored in your working directory. You'll need to commit these results into  the /trunk to finalize the merge.    
     
   

**SVN for LPSHUB Project:-**

Scenario: 1



Scenario: 2

****

**References:-**

<https://ariejan.net/2006/11/24/svn-how-to-structure-your-repository/>

Test SVN all features

<https://192.168.111.22:8443/svn/Proj4TestSVN/>

<http://tortoisesvn.net/docs/release/TortoiseSVN_en/>

<http://svnbook.red-bean.com/en/1.7/svn-> book.pdf#page=29&zoom=auto,37.519,445.669

<https://192.168.111.25:8443/svn/lpshub/trunk>

username :Mathan

pwd: m?qcp34Au0