SVN - LIFE CYCLE

The life cycle of a Version Control System is discussed in this chapter. In later chapters, we will see the Subversion command for each operation.

Create Repository:

The repository is a central place where developers store all their work. Repository not only stores files, but also the history about changes. Which means it maintains a history of the changes made in the files.

The 'create' operation is used to create a new repository. Most of the times this operation is done only once. When you create a new repository, your VCS will expect you to say something to identify it, such as where you want it to be created, or what name should be given to the repository.

Checkout

'Checkout' operation is used to create a working copy from the repository. Working copy is a private workplace where developers do their changes, and later on, submit these changes to the repository.

Update

As the name suggests, 'update' operation is used to update working copy. This operation synchronizes the working copy with the repository. As repository is shared by all the teams other developers can commit their changes and your working copy becomes older.

Let us suppose *Tom* and *Jerry* are the two developers working on a project. Both check out the latest version from the repository and start working. At this point, their working copies are completely synchronized with the repository. *Jerry* completes his work very efficiently and commits his changes to the repository.

Now *Tom's* working copy is out of date. Update operation will pull *Jerry's* latest changes from the repository and will update *Tom's* working copy.

Perform Changes

After the checkout, one can do various operations to perform changes. Edit is the most common operation. One can edit the existing file to add/remove contents from the file.

One can add files/directories. But immediately these files/directories do not become a part of the repository, instead they are added to the pending change-list and become a part of the repository after the commit operation.

Similarly one can delete files/directories. Delete operation immediately deletes file from the working copy, but actual deletion of the file is added to the pending change-list and changes are made to the repository after the commit operation.

'Rename' operation changes the name of the file/directory. 'Move' operation is used to move files/directories from one place to another in a repository tree.

Review Changes

When you check out the working copy or update the working copy, then your working copy is completely synchronized with the repository. But as you do changes to your working copy, it becomes newer than the repository. And it is a good practice to review your changes before the 'commit' operation.

'Status' operation lists the modifications that have been made to the working copy. As we have mentioned before, whenever you do changes in the working copy all these changes become a part of the pending change-list. And the 'status' operation is used to see the pending change-list.

'Status' operation only provides a list of changes but not the details about them. One can use *diff* operation to view the details of the modifications that have been made to the working copy.

Fix Mistakes

Let us suppose one has made changes to his working copy, but now, he wants to throw away these changes. In this situation, 'revert' operation will help.

Revert operation reverts the modifications that have been made to the working copy. It is possible to revert one or more files/directories. Also it is possible to revert the whole working copy. In this case, the 'revert' operation will destroy the pending change-list and will bring the working copy back to its original state.

Resolve Conflicts:

Conflicts can occur at the time of merging. 'Merge' operation automatically handles everything that can be done safely. Everything else is considered as conflict. For example, "*hello.c*" file was modified in branch and deleted in another branch. Such a situation requires a person to make the decision. The 'resolve' operation is used to help the user figure out things and to inform VCS about the ways of handling the conflicts.

Commit Changes

'Commit' operation is used to apply changes from the working copy to the repository. This operation modifies the repository and other developers can see these changes by updating their working copy.

Before commit, one has to add files/directories to the pending change-list. This is the place where changes wait to be committed. With commit, we usually provide a log message to explain why someone made changes. This log message becomes a part of the history of the repository. Commit is an atomic operation, which means either the entire commit succeeds or it is rolled back. Users never see half-finished commit.