

SUBVERSION

Subversion software is also known as SVN. It is an open source version control system. Through subversion we can look at the previous version of the file and track the changes over time.

There are two types of Version Control System:

- **Centralized Version Control System(CSCV):** There is a single central server that stores all the versions.
- **Distributed Version Control System(DVSC):** Each user have the copy of full repository.

Why do we use Version Control?

- To track all the changes and keep the history.
- We can rollback to the previous version when needed.
- We can merge new features.

Installing & Setting up SVN on windows:

1. Download and install SVN ([TortoiseSVN](#))
2. After the installation restart your system.
3. Verify the installation.

svn --version

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>svn --version
svn, version 1.14.5 (r1922182)
   compiled Nov 30 2024, 08:20:48 on x86-microsoft-windows

Copyright (C) 2024 The Apache Software Foundation.
This software consists of contributions made by many people;
see the NOTICE file for more information.
Subversion is open source software, see http://subversion.apache.org/

The following repository access (RA) modules are available:

* ra_svn : Module for accessing a repository using the svn network protocol.
  - with Cyrus SASL authentication
  - handles 'svn' scheme
* ra_local : Module for accessing a repository on local disk.
  - handles 'file' scheme
* ra_serf : Module for accessing a repository via WebDAV protocol using serf.
  - using serf 1.3.10 (compiled with 1.3.10)
  - handles 'http' scheme
  - handles 'https' scheme

The following authentication credential caches are available:

* Wincrypt cache in C:\Users\anugy\AppData\Roaming\Subversion

C:\Windows\System32>
```

SVN Commands:

Step 1: Initialize the repository.

svnadmin create ~/svn_repo/my_project

```
C:\Windows\System32>svnadmin create D:\Anugya_repo
```

Step 2: Checkout the repository

svn checkout file:///path/to/svn_repo/my_project

```
C:\Windows\System32>svn checkout file:///D:\Anugya_repo Anugya_project
Checked out revision 0.
```

Step 3: Add files to the directory

- Navigate to your working directory.
- Add new files.
- Commit the changes.

cd my_project

svn add file.txt

svn commit -m "Added file.txt"

```
C:\Windows\System32>cd Anugya_project

C:\Windows\System32\Anugya_project>echo "Hello SVN" > Anugya.txt

C:\Windows\System32\Anugya_project>svn add Anugya_file.txt
svn: warning: W155010: 'C:\Windows\System32\Anugya_project\Anugya_file.txt' not found
svn: E200009: Could not add all targets because some targets don't exist
svn: E200009: Illegal target for the requested operation

C:\Windows\System32\Anugya_project>svn add Anugya.txt
A
    Anugya.txt

C:\Windows\System32\Anugya_project>svn commit -m "Initial commit by Anugya"
Adding
    Anugya.txt
Transmitting file data .done
Committing transaction...
Committed revision 1.
```

Step 4: Update your working copy and view the logs

svn update

svn log

```
C:\Windows\System32\Anugya_project>svn update
Updating '.':
At revision 1.
```

```
C:\Windows\System32\Anugya_project>svn log
-----
r1 | anugy | 2025-02-16 13:36:50 +0530 (Sun, 16 Feb 2025) | 1 line
Initial commit by Anugya
-----
```

Step 5: Reverting the changes

svn revert file.txt

```
C:\Windows\System32\Anugya_project>svn revert Anugya.txt
```

Step 6: Creating the branch and merge the changes.

svn copy file:///C:/svn_repos/my_repo/trunk

file:///C:/svn_repos/my_repo/branches/faeature-branch -m "Creating feature branch"

svn merge file:///C:/svn_repos/my_repo/branches/faeature-branch

```
C:\Windows\System32\Anugya_project>svn mkdir file:///D:/Anugya_repo/branches -m "Creating Branches Directory"
Committing transaction...
Committed revision 2.
```

```
C:\Windows\System32\Anugya_project>svn mkdir file:///D:/Anugya_repo/trunk -m "Creating trunk directory"
Committing transaction...
Committed revision 3.
```

```
C:\Windows\System32\Anugya_project>svn copy file:///D:/Anugya_repo/trunk file:///D:/Anugya_repo/new-features -m "Creating a branch by Anugya"
Committing transaction...
Committed revision 4.
```

MERCURIAL (HG)

Mercurial is a distributed version control system (DVCS) designed for efficient handling of the projects of all sizes. The functionality of Mercurial is similar as Git but it emphasizes simplicity and ease of use. Mercurial is written in python and it is known for its intuitive commands, robust performance and cross-platform compatibility.

Features of Mercurial:

- **Distributed Version Control:** Every developer has a fully copy of the repository, enabling offline work and independent branching.
- **Lightweight and Fast:** Efficient handling of large projects and binary files.
- **Cross-Platform:** It works on Windows, macOS and Linux.
- **Extensible:** It supports plugins for additional functionality.
- **Simple and Intuitive Commands:** Commands are easy to learn and use, with a consistent syntax.

Installing & setting up Mercurial on Windows:

1. Download and install Mercurial ([TortoiseHg](#))
2. After the installation restart your system.
3. Verify the installation.

hg --version

```
C:\Windows\System32>hg --version
Mercurial Distributed SCM (version 6.5.1)
(see https://mercurial-scm.org for more information)

Copyright (C) 2005-2023 Olivia Mackall and others
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Mercurial Commands

Step 1: Creating and Initializing the Repository

hg init my-hg-repo

```
C:\Windows\System32>hg init Anugya_hg_repo  
C:\Windows\System32>cd Anugya_hg_repo  
C:\Windows\System32\Anugya_hg_repo>
```

Step 2: Adding files and committing the files

hg add file.txt

hg commit -m "Added newfile.txt"

```
C:\Windows\System32\Anugya_hg_repo>echo "Hello Mercurial" > Anugya_file.txt  
C:\Windows\System32\Anugya_hg_repo>hg add Anugya_file.txt  
C:\Windows\System32\Anugya_hg_repo>
```

Step 3: Cloning, Updating and Reverting

```
C:\Windows\System32\Anugya_hg_repo>hg commit -m "Initial commit by Anugya"  
C:\Windows\System32\Anugya_hg_repo>
```

hg clone https://example.com/repo

hg pull

hg update

hg log

```
C:\Windows\System32\Anugya_hg_repo>hg clone "C:\Windows\System32\Anugya_hg_repo" "C:\Windows\System32\Anugya_hg_clone"  
updating to branch default  
1 files updated, 0 files merged, 0 files removed, 0 files unresolved  
  
C:\Windows\System32\Anugya_hg_repo>hg pull "C:\Windows\System32\Anugya_hg_repo"  
pulling from C:\Windows\System32\Anugya_hg_repo  
searching for changes  
no changes found  
1 local changesets published  
  
C:\Windows\System32\Anugya_hg_repo>hg update  
0 files updated, 0 files merged, 0 files removed, 0 files unresolved
```

```
C:\Windows\System32\Anugya_hg_repo>hg log
changeset:  0:47f66e680ff5
tag:        tip
user:       Anugya Agarwal <anugyaagr3032@gmail.com>
date:       Sun Feb 16 14:17:43 2025 +0530
summary:    Initial commit by Anugya
```

```
C:\Windows\System32\Anugya_hg_repo>
```

Step 4: Branching and Merging

hg branch new-feature

hg merge

```
C:\Windows\System32\Anugya_hg_repo>hg branch Anugya_feature
marked working directory as branch Anugya_feature
(branches are permanent and global, did you want a bookmark?)

C:\Windows\System32\Anugya_hg_repo>hg commit -m "Creating a new feature branch by Anugya"

C:\Windows\System32\Anugya_hg_repo>echo "This is an update from Anugya-feature branch" >> Anugya_file.txt

C:\Windows\System32\Anugya_hg_repo>hg update default
0 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Windows\System32\Anugya_hg_repo>hg merge Anugya_feature
0 files updated, 0 files merged, 0 files removed, 0 files unresolved
(branch merge, don't forget to commit)

C:\Windows\System32\Anugya_hg_repo>

C:\Windows\System32\Anugya_hg_repo>hg commit -m "Merged Anugya-feature into default"

C:\Windows\System32\Anugya_hg_repo>
```