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.

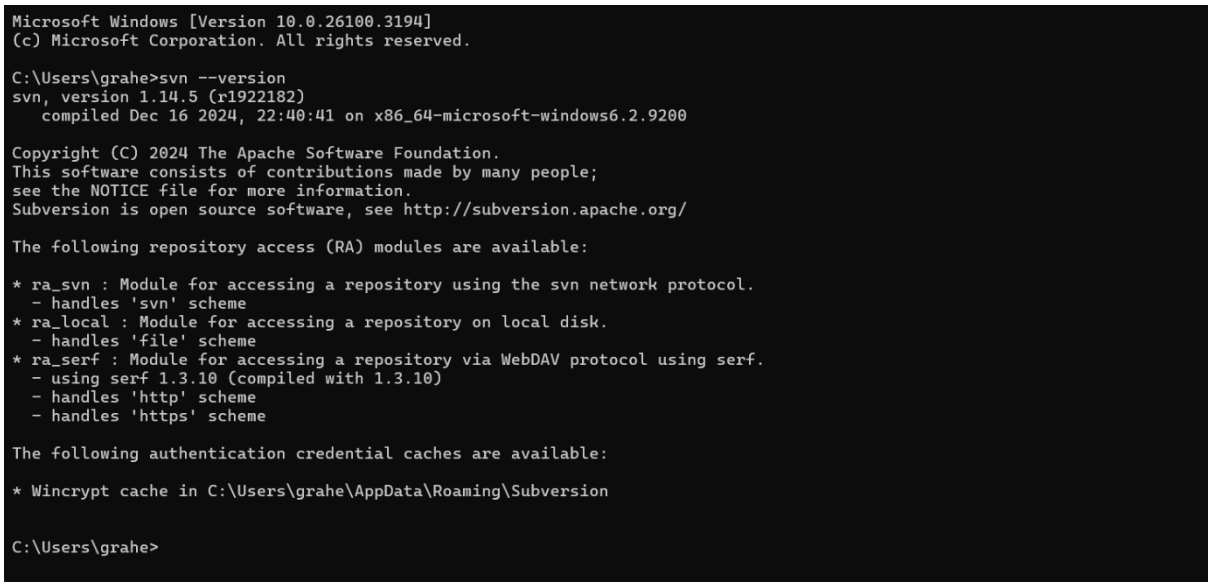
Why do we use Version Control?

- 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



SVN Commands:

Step 1: Initialize the repository.

svnadmin create ~/svn_repo/my_project

```
C:\Users\grahe>svnadmin create G:\graheet_repo
```

Step 2: Checkout the repository

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

```
C:\Users\grahe>svn checkout file:///G:/graheet_repo graheet-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:\Users\grahe>cd graheet-project

C:\Users\grahe\graheet-project>echo "Hello SVN" > Graheet_file.txt

C:\Users\grahe\graheet-project>svn add Graheet_file.txt
A         Graheet_file.txt

C:\Users\grahe\graheet-project>svn commit -m "Initial commit by Graheet"
Adding         Graheet_file.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:\Users\grahe\graheet-project>svn update
Updating '.':
At revision 1.

C:\Users\grahe\graheet-project>svn log
------------------------------------------------------------------------
r1 | grahe | 2025-02-15 19:04:15 +0530 (Sat, 15 Feb 2025) | 1 line

Initial commit by Graheet
------------------------------------------------------------------------
```

Step 5: Reverting the changes

svn revert file.txt

```
C:\Users\grahe\graheet-project>svn revert graheet_file.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:\Users\grahe\graheet-project>svn mkdir file:///G:/graheet_repo/trunk -m "Creating trunk"
Committing transaction...
Committed revision 2.
```

```
C:\Users\grahe\graheet-project>svn mkdir file:///G:/graheet_repo/branches -m "Creating branches directory"
Committing transaction...
Committed revision 3.
```

```
C:\Users\grahe\graheet-project>svn mkdir file:///G:/graheet_repo/branches/new-feature -m "Creating a branch by Graheet"
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:\Users\grahe>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:\Users\grahe>hg init Graheet_hg_repo

C:\Users\grahe>cd Graheet_hg_repo

C:\Users\grahe\Graheet_hg_repo>
```

Step 2: Adding files and committing the files

hg add file.txt

hg commit -m “Added newfile.txt”

```
C:\Users\grahe\Graheet_hg_repo>echo "Hello Mercurial" > Graheet_file.txt

C:\Users\grahe\Graheet_hg_repo>hg add Graheet_file.txt

C:\Users\grahe\Graheet_hg_repo>hg config --edit
```

```
C:\Users\grahe\Graheet_hg_repo>hg config --edit

C:\Users\grahe\Graheet_hg_repo>hg commit -m "Initial Commit by Graheet"

C:\Users\grahe\Graheet_hg_repo>
```

Step 3: Cloning, Updating and Reverting

hg clone https://example.com/repo

hg pull

hg update

hg log

```
C:\Users\grahe\Graheet_hg_repo>hg pull "C:/Users/grahe/Graheet_hg_repo"
pulling from C:/Users/grahe/Graheet_hg_repo
searching for changes
no changes found
1 local changesets published

C:\Users\grahe\Graheet_hg_repo>hg update
0 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Users\grahe\Graheet_hg_repo>hg log
changeset:   0:c980946d4e25
tag:         tip
user:        Graheet
date:        Sat Feb 15 23:52:50 2025 +0530
summary:     Initial Commit by Graheet
```

```
C:\Users\grahe\Graheet_hg_repo>hg revert Graheet_file.txt
no changes needed to Graheet_file.txt
```

Step 4: Branching and Merging

hg branch new-feature

hg merge

```
C:\Users\grahe\Graheet_hg_repo>hg branch Graheet-feature
marked working directory as branch Graheet-feature
(branches are permanent and global, did you want a bookmark?)

C:\Users\grahe\Graheet_hg_repo>hg commit -m "Creating a new feature branch by Graheet"

C:\Users\grahe\Graheet_hg_repo>echo "This ia an update from Graheet-feature branch" >> Graheet_file.txt

C:\Users\grahe\Graheet_hg_repo>hg commit -m "Updated Graheet_file.txt  in Graheet-feature branch"

C:\Users\grahe\Graheet_hg_repo>hg update default
1 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Users\grahe\Graheet_hg_repo>hg merge Graheet-feature
1 files updated, 0 files merged, 0 files removed, 0 files unresolved
(branch merge, don't forget to commit)

C:\Users\grahe\Graheet_hg_repo>hg commit -m "Merged Graheet-feature into default"
```