



## **DEVOPS**

Roll No.: R2142220334

SAP ID:500102094

Batch: B1

Semester: VI

Submitted by: Janhvi Jaiswal

Submitted to: Prof. Prateek Raj Gautam

# SUBVERSION

## Subversion (SVN)

Subversion (SVN) is an open-source version control system (VCS) for managing changes to files over a period of time, enabling collaboration, and keeping track of multiple versions of a project.

## Types of Version Control Systems

### Centralized Version Control (CVCS):

All versions are stored on a central server. Users check out the current version to work on and commit changes to the server (e.g., SVN).

### Distributed Version Control (DVCS):

Each user keeps a full copy of the repository, which facilitates offline work and solo branching (e.g., Git).

## Why Use Version Control?

1. **Track Changes:** Maintains a historical account of all the changes.
2. **Revert Versions:** Roll back versions easily if required.
3. **Collaboration:** Can have several people working on one project at once without any conflict.
4. **Branching & Merging:** Develop on features independently and merge them without affecting the master project.

## Installing SVN on Windows

### Download & Install TortoiseSVN

Go to TortoiseSVN and download the version that is suitable for your operating system.

### Restart System:

Restart your computer after installation for seamless integration with Windows Explorer.

### Verify Installation:

```
svn --version
```

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

C:\Users\JANHVI>svn --version
svn, version 1.14.5 (r1922182)
  compiled Dec 16 2024, 22:40:41 on x86_64-microsoft-windows6.2.9200

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.
  - 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\JANHVI\AppData\Roaming\Subversion
```

## SVN Commands:

### Step 1: Initialize the repository.

**svnadmin create ~/svn\_repo/my\_project**

```
C:\Users\JANHVI>svnadmin create C:\janhvi_repo
```

### Step 2: Checkout the repository

**svn checkout [file:///path/to/svn\\_repo/my\\_project](#)**

```
C:\Users\JANHVI>svn checkout file:///C:/janhvi_repo janhvi_project
Checked out revision 0.
```

### Step 3: Add files to the directory

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

**svn add file.txt**

**svn commit -m "Added file.txt"**

```

C:\Users\JANHVI>cd janhvi_project

C:\Users\JANHVI\janhvi_project>echo "Hello SVN" > janhvi_file.txt

C:\Users\JANHVI\janhvi_project>svn add janhvi_file.txt
A
    janhvi_file.txt

C:\Users\JANHVI\janhvi_project>svn commit -m "Initial commit by Janhvi"
Adding          janhvi_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\JANHVI\janhvi_project>svn update
Updating '.':
At revision 1.

C:\Users\JANHVI\janhvi_project>svn log
-----
r1 | JANHVI | 2025-02-16 15:10:02 +0530 (Sun, 16 Feb 2025) | 1 line

Initial commit by Janhvi
-----

```

#### Step 5: Reverting the changes

svn revert file.txt

```

C:\Users\JANHVI\janhvi_project>svn revert janhvi_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/trunk) [file:///C:/svn\\_repos/my\\_repo/branches/feature-branch](file:///C:/svn_repos/my_repo/branches/feature-branch) -m "Creating feature branch"

svn merge [file:///C:/svn\\_repos/my\\_repo/branches/feature-branch](file:///C:/svn_repos/my_repo/branches/feature-branch)

```

C:\Users\JANHVI\janhvi_project>svn mkdir file:///C:/janhvi_repo/trunk -m "Creating trunk"
svn mkdir file:///C:/janhvi_repo/
ing trunk"
Committing transaction...
Committed revision 2.

```

```

C:\Users\JANHVI\janhvi_project>svn mkdir file:///C:/janhvi_repo/branches -m "Creating branches directory"
Committing transaction...
Committed revision 3.

```

```

C:\Users\JANHVI\janhvi_project>svn mkdir file:///C:/janhvi_repo/branches/new-feature -m "Creating a branch by Janhvi"
Committing transaction...
Committed revision 4.

```

# MERCURIAL (HG)

Mercurial is a distributed version control system (DVCS) that is meant for effective management of the projects of all sizes. Mercurial's functionality is similar to Git but it focuses on simplicity and ease of use. Mercurial is implemented in python and it is famous for its easy-to-use commands, strong performance and cross-platform support.

## Features of Mercurial:

- Distributed Version Control:** Each developer has a complete copy of the repository, supporting offline development and independent branching.
- Lightweight and Fast:** Optimal management of large projects and binary files.
- Cross-Platform:** It supports Windows, macOS and Linux.
- Extensible:** Plugins for extra functionality are supported.
- Simple and Intuitive Commands:** Easy-to-learn, easy-to-use commands with uniform syntax.

## Installing & setting up Mercurial on Windows:

1. Download and install Mercurial (TortoiseHg)
2. After installing restart your machine.
3. Check the installation.

hg --version

```
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jitendra Kumar>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\Jitendra Kumar>hg init Anugrah_hg_repo
C:\Users\Jitendra Kumar>cd Anugrah_hg_repo
C:\Users\Jitendra Kumar\Anugrah_hg_repo>
```

### Step 2: Adding files and committing the files

**hg add file.txt**

**hg commit -m "Added newfile.txt"**

```
C:\Users\Jitendra Kumar\Anugrah_hg_repo>echo "Hello Mercurial" > Anugrah_file.txt
C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg add Anugrah_file.txt
```

```
C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg commit -m "Initial commit by Anugrah"
nothing to commit, working tree clean
```

### Step 3: Cloning, Updating and Reverting

**hg clone <https://example.com/repo>**

**hg pull**

**hg update**

**hg log**

```
C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg clone "C:/Users/Jitendra Kumar/Anugrah_hg_repo" "C:/Users/Jitendra Kumar/Anugrah_hg_clone"
updating to branch default
1 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg pull
pulling from default
abort: repository default not found

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg pull "C:/Users/Jitendra Kumar/Anugrah_hg_repo"
pulling from C:/Users/Jitendra Kumar/Anugrah_hg_repo
searching for changes
no changes found
1 local changesets published

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg update
0 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg log
changeset: 0:e23a57b6dfbd
tag:       tip
user:      Anugrah <anugrahkvs@gmail.com>
date:      Sat Feb 15 19:21:53 2025 +0530
summary:   Initial Commit by Anugrah

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg revert Anugrah_file.txt
no changes needed to Anugrah_file.txt
```

### Step 4: Branching and Merging

**hg branch new-feature**

**hg merge**

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

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg commit -m "Creating a new feature branch by Anugrah"

C:\Users\Jitendra Kumar\Anugrah_hg_repo>echo "This ia an update from Anugrah-feature branch" >> Anugrah_file.txt

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg commit -m "Updated Anugrah_file.txt in Anugrah-feature branch"

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg update default
1 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg merge Anugrah-feature
1 files updated, 0 files merged, 0 files removed, 0 files unresolved
(branch merge, don't forget to commit)

C:\Users\Jitendra Kumar\Anugrah_hg_repo>hg commit -m "Merged Anugrah-feature into default"
```