UPES

UNIVERSITY OF TOMORROW

DevOps
Assignment-1

Roll No.: R2142220302

SAP ID:500102207

Batch: B2

Semester: VI

Submitted by: Abhinav Kukreti

Submitted to: Dr. Prateek Raj Gautam

VERSION CONTROL ASSIGNMENT-1

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.

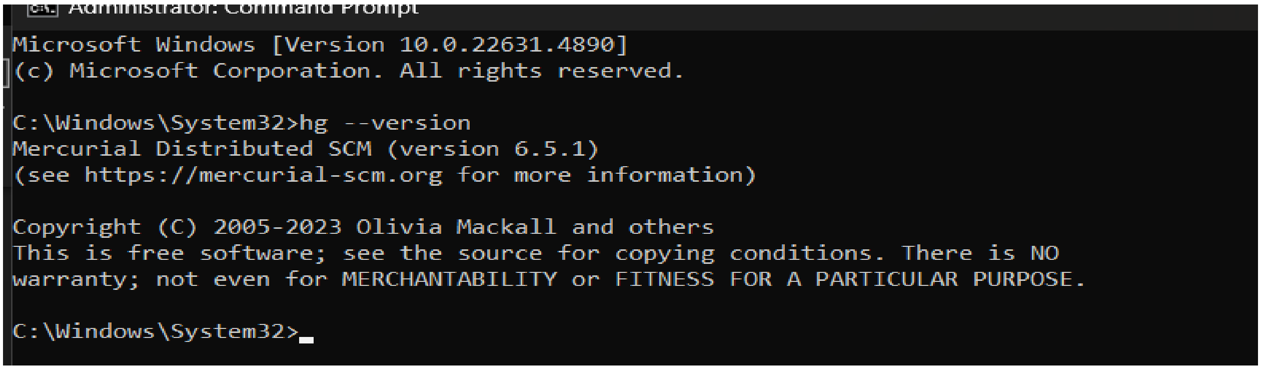
Features of Mercurial:

- **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

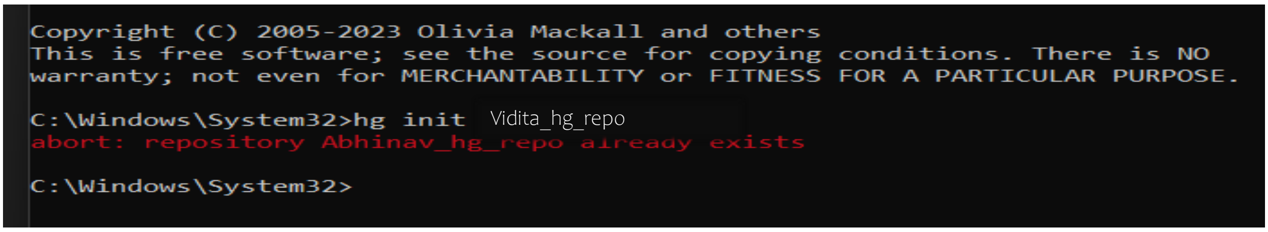


```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

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.

C:\Windows\System32>
```

Mercurial Commands

Step 1: Creating and Initializing the Repository

hg init my-hg-repo



```
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.

C:\Windows\System32>hg init  Vidita_hg_repo
abort: repository Abhinav_hg_repo already exists

C:\Windows\System32>
```

S

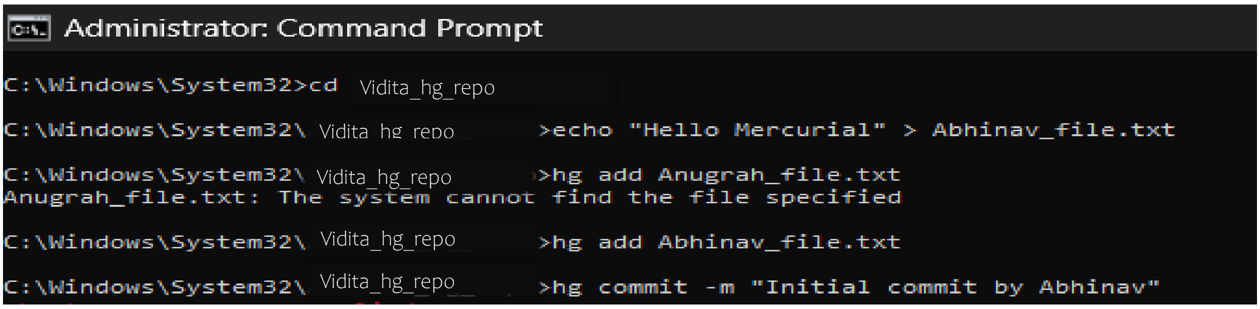
t

e

p 2: Adding files and committing the files hg

add file.txt

hg commit -m “Added newfile.txt”



Administrator: Command Prompt

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

C:\Windows\System32\ Vidita_hg_repo      >echo "Hello Mercurial" > Abhinav_file.txt

C:\Windows\System32\ Vidita_hg_repo      >hg add Anugrah_file.txt
Anugrah_file.txt: The system cannot find the file specified

C:\Windows\System32\ Vidita_hg_repo      >hg add Abhinav_file.txt

C:\Windows\System32\ Vidita_hg_repo      >hg commit -m "Initial commit by Abhinav"
```

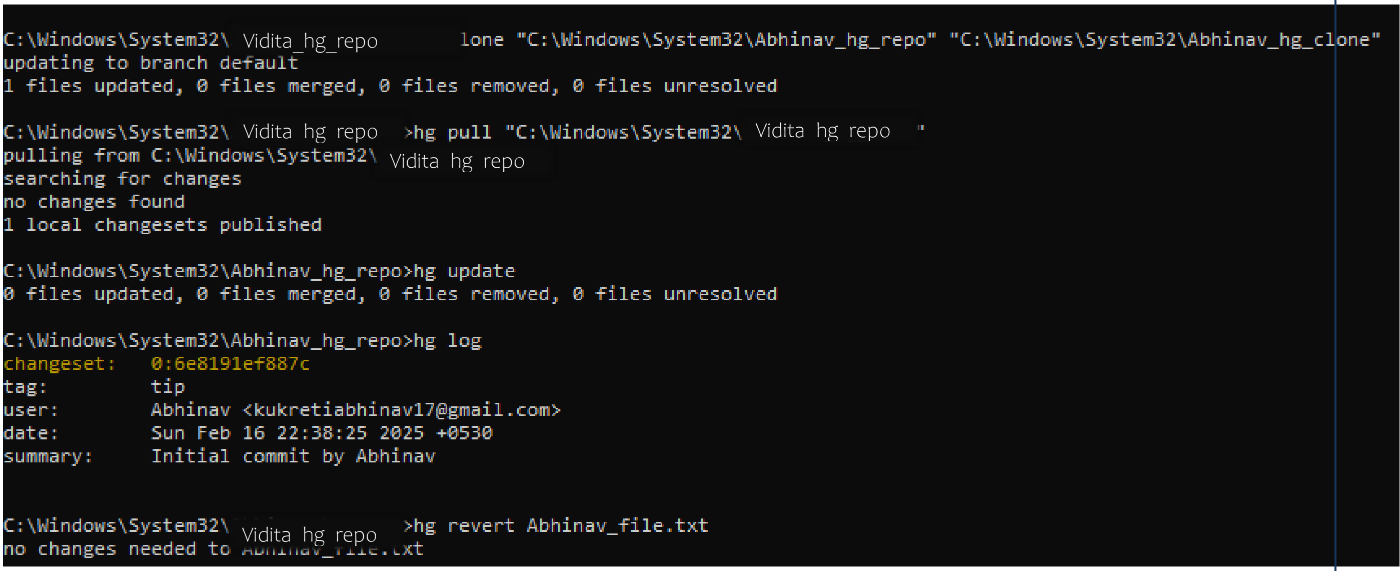
Step 3: Cloning, Updating and Reverting

hg clone https://example.com/repo

hg pull

hg update

hg log



```
C:\Windows\System32\ Vidita_hg_repo        lone "C:\Windows\System32\Abhinav_hg_repo" "C:\Windows\System32\Abhinav_hg_clone"
updating to branch default
1 files updated, 0 files merged, 0 files removed, 0 files unresolved

C:\Windows\System32\ Vidita hg repo  >hg pull "C:\Windows\System32\ Vidita hg repo  "
pulling from C:\Windows\System32\ Vidita hg repo
searching for changes
no changes found
1 local changesets published

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

C:\Windows\System32\Abhinav_hg_repo>hg log
changeset:   0:6e8191ef887c
tag:         tip
user:        Abhinav <kukretiabhinav17@gmail.com>
date:        Sun Feb 16 22:38:25 2025 +0530
summary:     Initial commit by Abhinav


C:\Windows\System32\ Vidita hg repo  >hg revert Abhinav_file.txt
no changes needed to Abhinav_file.txt
```

Step 4: Branching and Merging

```
hg branch new-feature
```

```
hg merge
```

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.

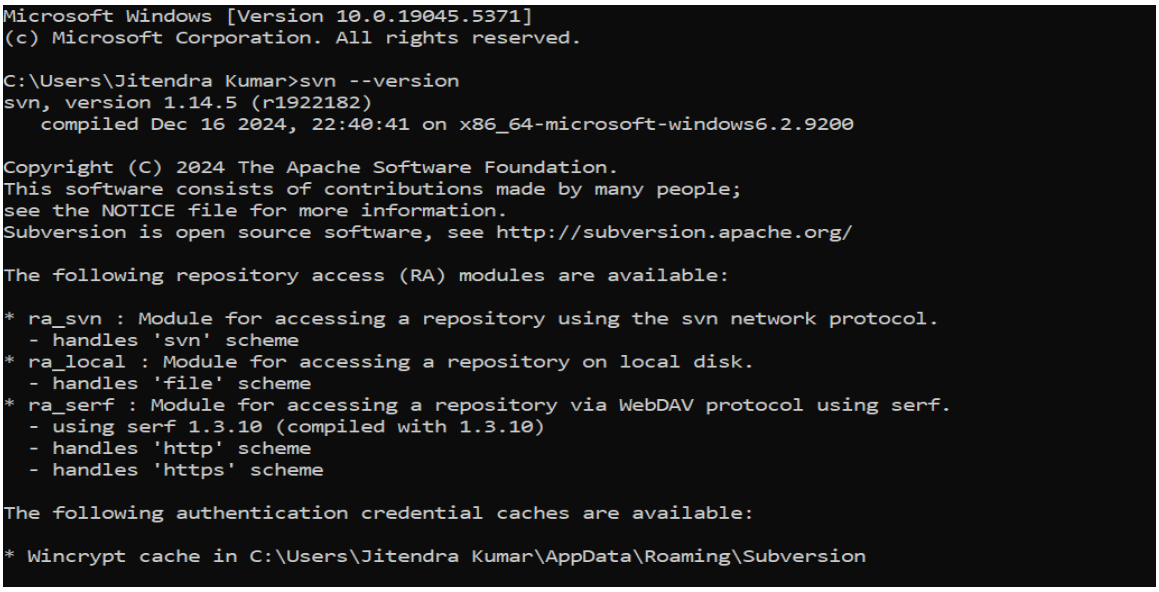
Why do we use Version Control?

- 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
```

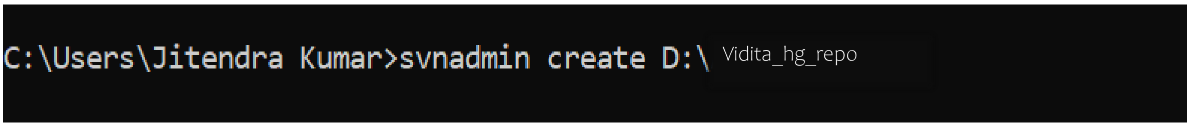


```
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jitendra Kumar>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\Jitendra Kumar\AppData\Roaming\Subversion
```

SVN Commands:

Step 1: Initialize the repository.

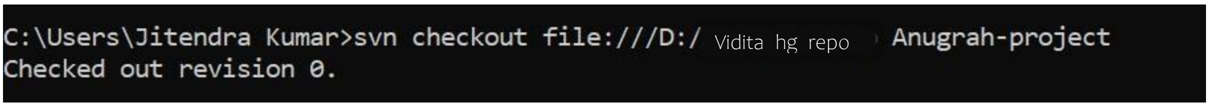
svnadmin create ~/svn_repo/my_project



```
C:\Users\Jitendra Kumar>svnadmin create D:\ Vidita_hg_repo
```

Step 2: Checkout the repository

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



```
C:\Users\Jitendra Kumar>svn checkout file:///D:/ Vidita hg repo  Anugrah-project
Checked out revision 0.
```

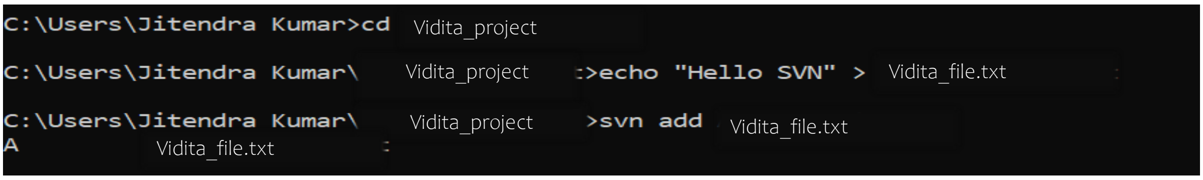
Step 3: Add files to the directory

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

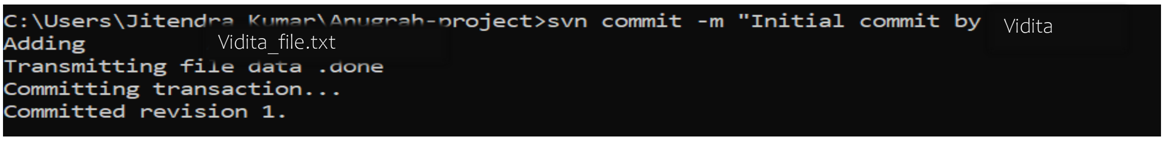
cd my_project

svn add file.txt

svn commit -m "Added file.txt"



```
C:\Users\Jitendra Kumar>cd  Vidita_project

C:\Users\Jitendra Kumar\  Vidita_project  >echo "Hello SVN" >  Vidita_file.txt

C:\Users\Jitendra Kumar\  Vidita_project  >svn add  Vidita_file.txt
A          Vidita_file.txt
```

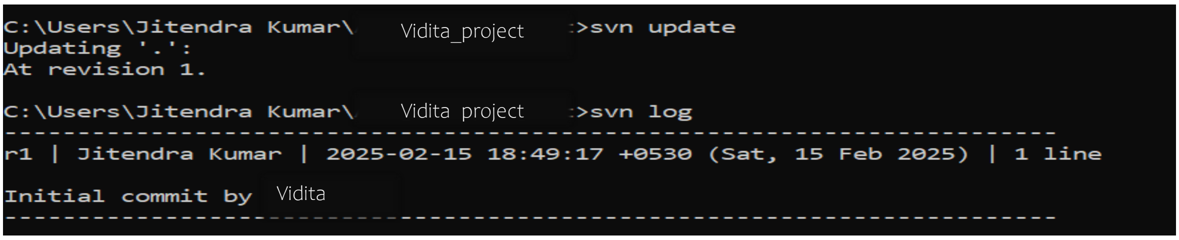


```
C:\Users\Jitendra Kumar\Anugrah-project>svn commit -m "Initial commit by  Vidita
Adding          Vidita_file.txt
Transmitting file data .done
Committing transaction...
Committed revision 1.
```

Step 4: Update your working copy and view the logs

svn update

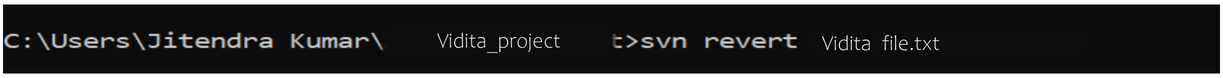
svn log



```
C:\Users\Jitendra Kumar\    Vidita_project    :>svn update
Updating '.':
At revision 1.

C:\Users\Jitendra Kumar\    Vidita_project    :>svn log
------------------------------------------------------------------------
r1 | Jitendra Kumar | 2025-02-15 18:49:17 +0530 (Sat, 15 Feb 2025) | 1 line

Initial commit by  Vidita
------------------------------------------------------------------------
```

Step 5: Reverting the changes

svn revert file.txt



```
C:\Users\Jitendra Kumar\    Vidita_project    t>svn revert  Vidita 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