# PAL DISTRIBUTED SOURCE CONTROL CLI TOOL WALKTHROUGH.

The following is an example of using the Pal CLI, showcasing the complete functionality of the distributed source control system.

To set up, I opened the Command Prompt (CMD) as an administrator in my Windows environment. I navigated to a directory of my choice and ran **`code .`** to initialize my IDE of choice, Visual Studio Code (VS Code), and opened the terminal within the IDE.

Next, I ran **`node -v`** and **`npm -v`** to confirm that Node.js and npm are installed and accessible in this location.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> node -v
v20.17.0
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> npm -v
10.9.2
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

Now we proceed to use our CLI tool, which exists as an npm package. We install it globally by running the command **`npm install -g @kennnyamai/pal-sc-cli`**. After the installation, we verify its success by running **`pal --version`** to confirm it has been installed correctly.

```
TERMINAL                                                    powershell

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> npm install -g @kennynyamai/pal-sc-cli

added 68 packages in 31s

9 packages are looking for funding
  run `npm fund` for details
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal --version
1.0.0
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

The first command we use is **`pal init`**, which initializes a repository in the current directory. Running this command prompts you to enter your name and email address, which will later be used as author details when making commits.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal init

Welcome to Pal! Let's set up your repository. 🚀

Enter your name: kennedy
Enter your email: kennynyamai73@gmail.com

[INFO] Preparing your repository...


[SUCCESS] Repository initialized successfully! 🎉
Location: C:\Users\KENNENDY MWENDWA\Documents\Source_Control\.pal

User information:
  Name:  kennedy
  Email: kennynyamai73@gmail.com

Next steps:
  1. Add files to your repository using `pal add <file>`
  2. Commit your changes with `pal commit -m "your message"`
  3. Check the status of your repository using `pal status`

For a complete list of commands, use `pal help`.
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

After successfully executing the **`pal init`** command, running the **ls** command (or **dir,** depending on your system) displays a newly created dot-prefixed subdirectory (visible only if hidden directories are enabled on your machine). This subdirectory contains the repository properties, equivalent to Git's internal objects.

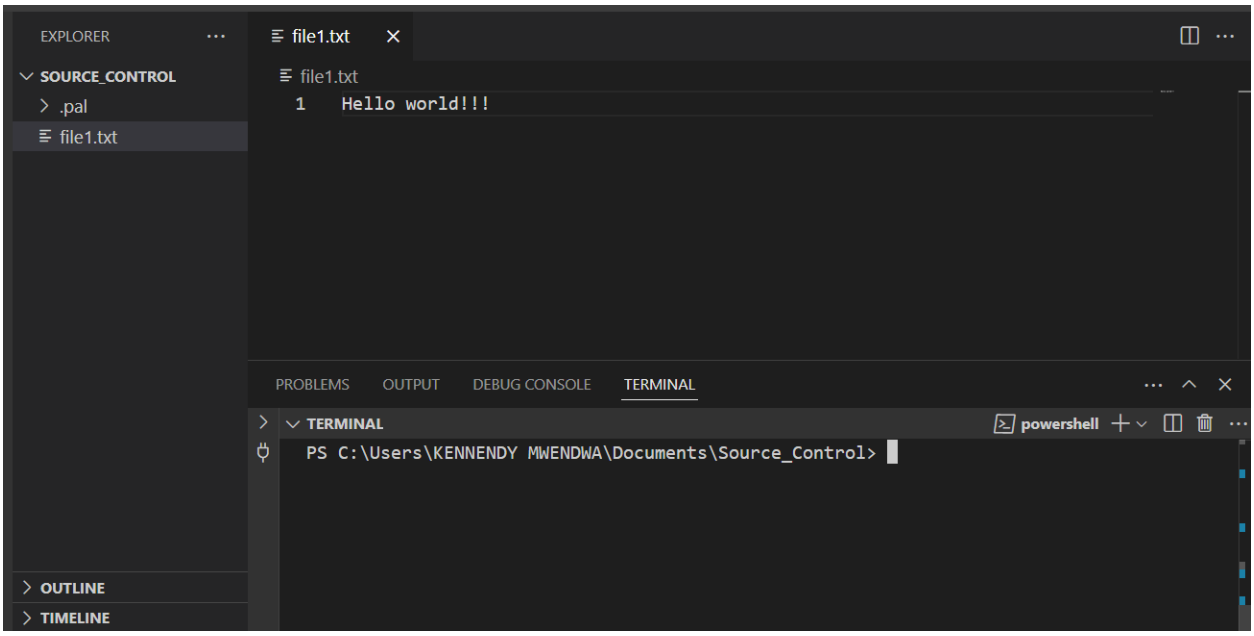
```
TERMINAL                                                    powershell

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> ls


    Directory: C:\Users\KENNENDY MWENDWA\Documents\Source_Control


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        12/12/2024   4:22 AM                .pal


PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

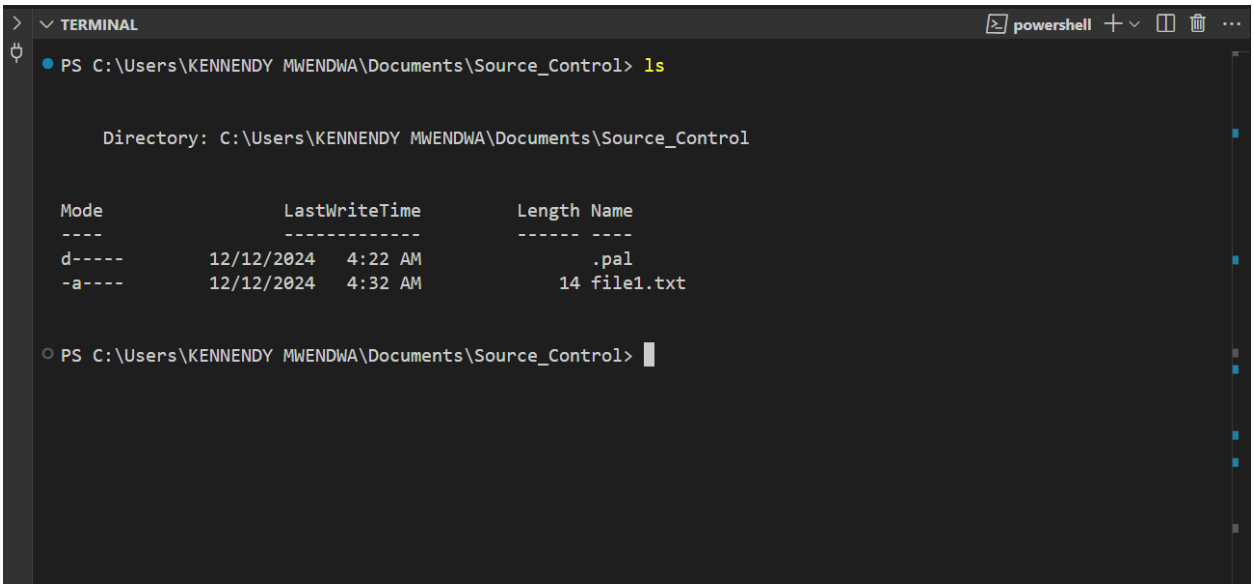
```
TERMINAL                                                    powershell - .pal
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> cd .pal
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control\.pal> ls


    Directory: C:\Users\KENNENDY MWENDWA\Documents\Source_Control\.pal


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        12/12/2024   4:22 AM                branches
d-----        12/12/2024   6:54 AM                objects
d-----        12/12/2024   5:51 AM                refs
-a----        12/12/2024   4:22 AM            119 config
-a----        12/12/2024   4:22 AM             59 description
-a----        12/12/2024   5:33 AM             27 HEAD
-a----        12/12/2024   6:52 AM            236 index


PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control\.pal>
```

Alternatively, you can run **`pal init <name_of_directory>`**, which creates a new directory with the specified name in the current location and initializes a repository within it.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal init test_dir

Welcome to Pal! Let's set up your repository. 🚀

Enter your name: kennedy
Enter your email: kennynyamai73@gmail.com

[INFO] Preparing your repository...


[SUCCESS] Repository initialized successfully! 🎉
Location: C:\Users\KENNENDY MWENDWA\Documents\Source_Control\test_dir\.pal

User information:
  Name:  kennedy
  Email: kennynyamai73@gmail.com

Next steps:
  1. Add files to your repository using `pal add <file>`
  2. Commit your changes with `pal commit -m "your message"`
  3. Check the status of your repository using `pal status`
```

Next, we move to the **`pal add`** command, which stages files. To demonstrate this, I first created a text file (**`file1.txt`**) at the root of my directory. Now my directory contains two entries: the .pal directory created earlier and file1.txt.



The **`touch`** command is not enabled on my terminal, but running touch file1.txt followed by **`echo "Hello world!!!" >> file1.txt`** achieves the same result. Once the file is created, we can stage it by running **`pal add file1.txt`**.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal add file1.txt

  📂 Repository located successfully.
  Adding files to the index...

  Index file does not exist, returning empty index.
  📄 Preparing files for staging...
  Index file does not exist, returning empty index.

  ✅ Index updated successfully.


  ✅ Files staged successfully:
    - file1.txt

  🎉 Use `pal commit` to save your changes.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

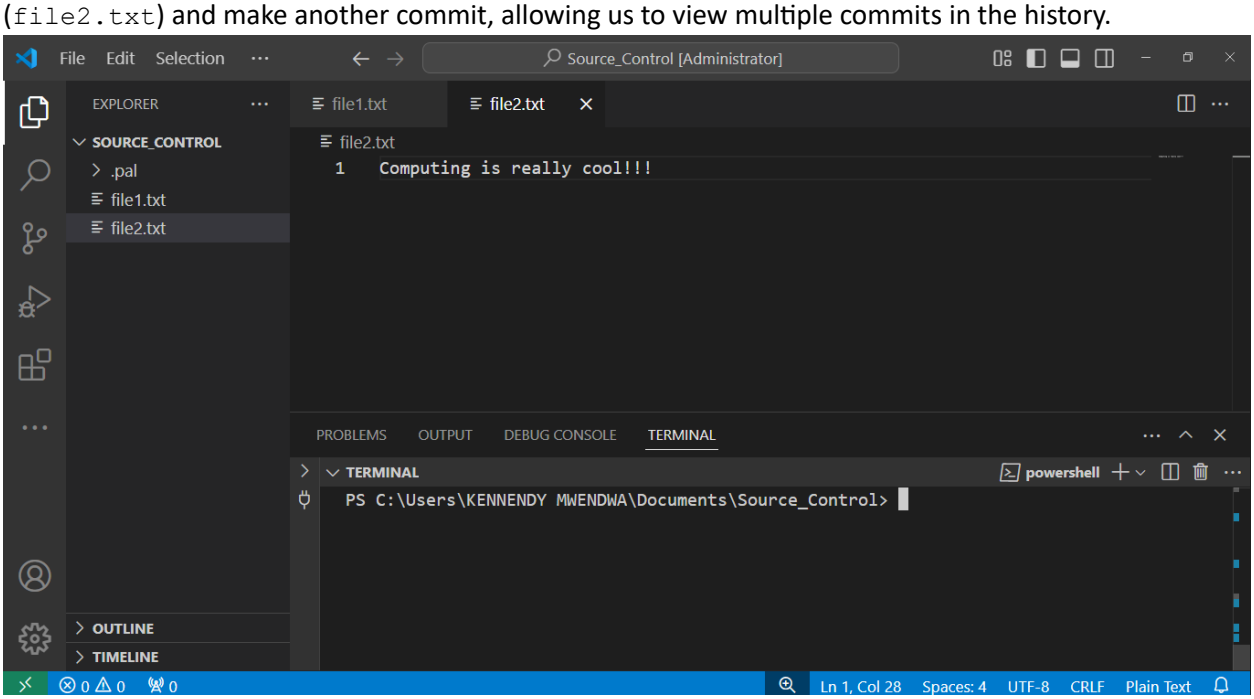
The **`pal ls-files`** command displays the files that have been staged.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal ls-files
[INFO] Found 1 staged file(s):
file1.txt
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

At this point, we can make our first commit by running **`pal commit`**. Optionally, you can include a custom commit message using **`pal commit -m "<custom commit message>"`**.

```
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal commit -m "initial commit"

[INFO] Generating tree object from the index...
[INFO] Fetching user details...
[INFO] Resolving parent commit...
[INFO] This is the first commit in the branch.
[INFO] Creating commit object...
Creating a new commit...
Commit created successfully with SHA: 0a3a7c28d5f5ae9be6586d1f81f1e5a4f767742a
[INFO] Updating branch reference...

[SUCCESS] Commit created successfully!
Commit SHA: 0a3a7c28d5f5ae9be6586d1f81f1e5a4f767742a

🎉 Use `pal log` to view your commit history.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

At this point, we can view our commit history by running **`pal log`**. To test this, I will stage another file (`file2.txt`) and make another commit, allowing us to view multiple commits in the history.



```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal ls-files
[INFO] Found 2 staged file(s):
file1.txt
file2.txt
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal commit -m "adding file2.txt"

[INFO] Generating tree object from the index...
[INFO] Fetching user details...
[INFO] Resolving parent commit...
[INFO] Creating commit object...
Creating a new commit...
Commit created successfully with SHA: c5dbca11a44ba90573dc2144a357a4e2feb7a0d9
[INFO] Updating branch reference...

[SUCCESS] Commit created successfully!
Commit SHA: c5dbca11a44ba90573dc2144a357a4e2feb7a0d9

🎉 Use `pal log` to view your commit history.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal log

📜 Commit history for branch starting at HEAD:

Commit: c5dbca11
Author: kennedy <kennynyamai73@gmail.com> 1733968457.095 180
Message: adding file2.txt
Parent: 0a3a7c28
------------------------------------------------
Commit: 0a3a7c28
Author: kennedy <kennynyamai73@gmail.com> 1733967710.417 180
Message: initial commit
------------------------------------------------

✅Log display completed successfully.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

The next functionality is branch management. Running **`pal branch`** displays a list of the existing branches. Currently, we have only the `master` branch, which was created automatically by the source control system when we made our first commit. The asterisk (*) indicates the active branch.
To create a new branch, we run **`pal branch <custom_branch_name>`**, and we can switch to this

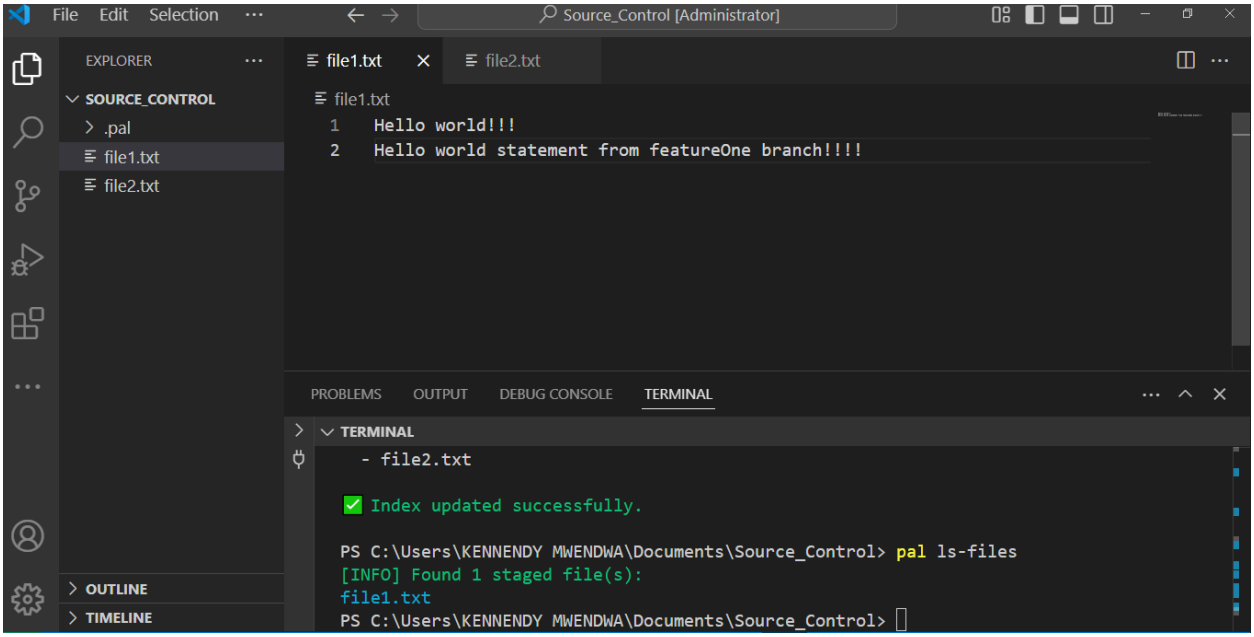
branch by running **`pal checkout <custom_branch_name>`**.

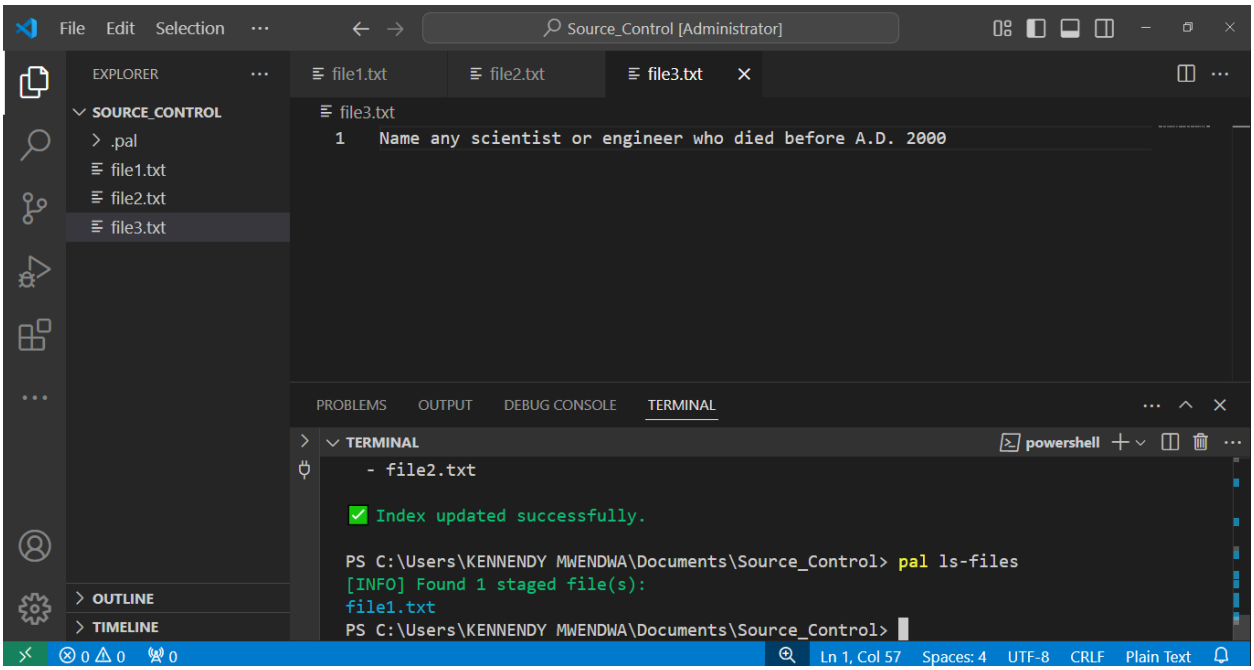
```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal branch

 Available branches:

  * master

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal branch featureOne
✓Branch 'featureOne' created successfully.
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal branch

 Available branches:

    featureOne
  * master

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal checkout featureOne
✓Switched to branch 'featureOne' successfully.
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

The next functionality we'll explore is diffing (comparing changes) between branches, specifically between the `master` branch and `featureOne`. To create conflicting changes between these two branches, we will perform the following steps:

1. Remove `file2.txt` from the staging area by running the `pal rm file2.txt` command (this command removes existing files from the staging area).
2. Reopen `file1.txt` and add another line to it.
3. Create a new text file called `file3.txt`, which does not exist in the `master` branch.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal ls-files
[INFO] Found 2 staged file(s):
file1.txt
file2.txt
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal rm file2.txt

⚠Removing paths from the index:
  - file2.txt

✅ Index updated successfully.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal ls-files
[INFO] Found 1 staged file(s):
file1.txt
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```


File Edit Selection
Source_Control [Administrator]
EXPLORER
SOURCE_CONTROL
.pal
file1.txt
file2.txt
file1.txt
file2.txt
1 Hello world!!!
2 Hello world statement from featureOne branch!!!!
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
TERMINAL
- file2.txt
Index updated successfully.
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal ls-files
[INFO] Found 1 staged file(s):
file1.txt
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
OUTLINE
TIMELINE


File Edit Selection
Source_Control [Administrator]
EXPLORER
SOURCE_CONTROL
.pal
file1.txt
file2.txt
file3.txt
file1.txt
file2.txt
file3.txt
1 Name any scientist or engineer who died before A.D. 2000
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
TERMINAL
powershell
- file2.txt
Index updated successfully.
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal ls-files
[INFO] Found 1 staged file(s):
file1.txt
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
OUTLINE
TIMELINE
Ln 1, Col 57 Spaces: 4 UTF-8 CRLF Plain Text

We will then stage these changes and commit them.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal add file1.txt file3.txt

  Repository located successfully.
Adding files to the index...

  Preparing files for staging...

Removing paths from the index:
  - file1.txt

  Index updated successfully.


  Files staged successfully:
  - file1.txt
  - file3.txt

  Use `pal commit` to save your changes.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal commit -m "changes on featureOne branch"

[INFO] Generating tree object from the index...
[INFO] Fetching user details...
[INFO] Resolving parent commit...
[INFO] Creating commit object...
Creating a new commit...
Commit created successfully with SHA: 199228bb576cd7c104974ef6e010ccb4023a0e35
[INFO] Updating branch reference...

[SUCCESS] Commit created successfully!
Commit SHA: 199228bb576cd7c104974ef6e010ccb4023a0e35

  Use `pal log` to view your commit history.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

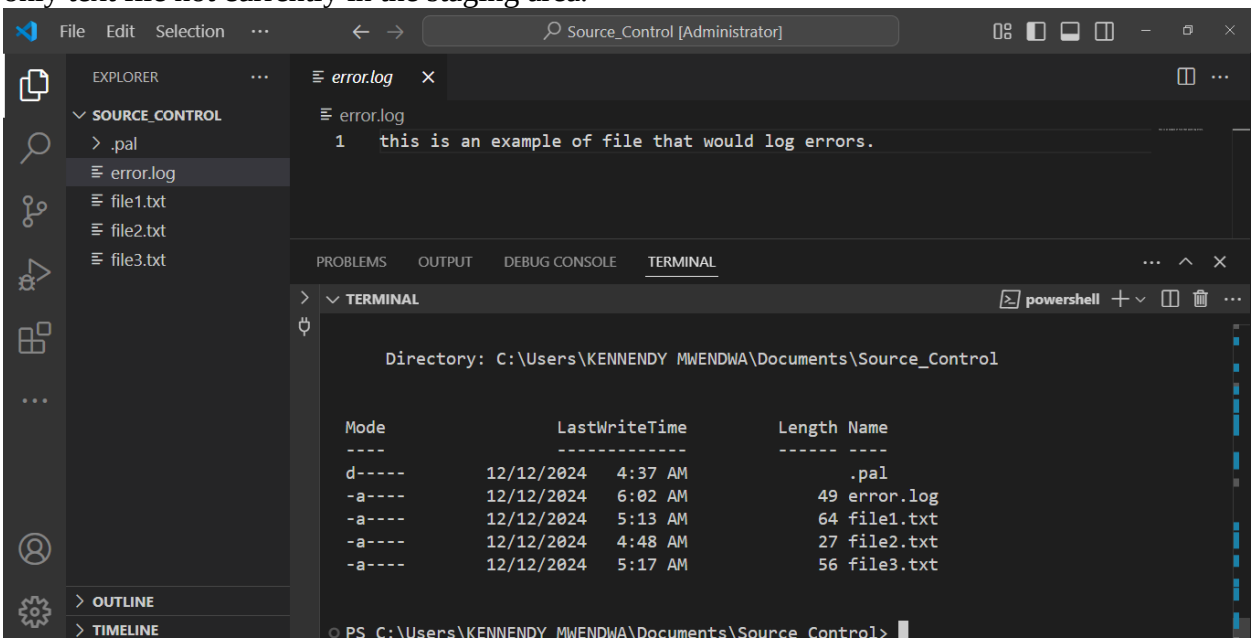
Now, we can view the differences between the two branches by running `pal diff <first_branch> <second_branch>`. This command will detect and display any conflicting changes between the branches.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal diff master featureOne

  Added files:
    + file3.txt

  Deleted files:
    - file2.txt

  Modified files:
    * file1.txt

  Conflicting files:
    ! file1.txt
      Conflict between SHA-1s: 19b67fb1fe22dd6c5860ed779ac9cd505f9d178a and 1c906c6e878a8a71bd8442cb86ea7d2d5b5a
733e

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

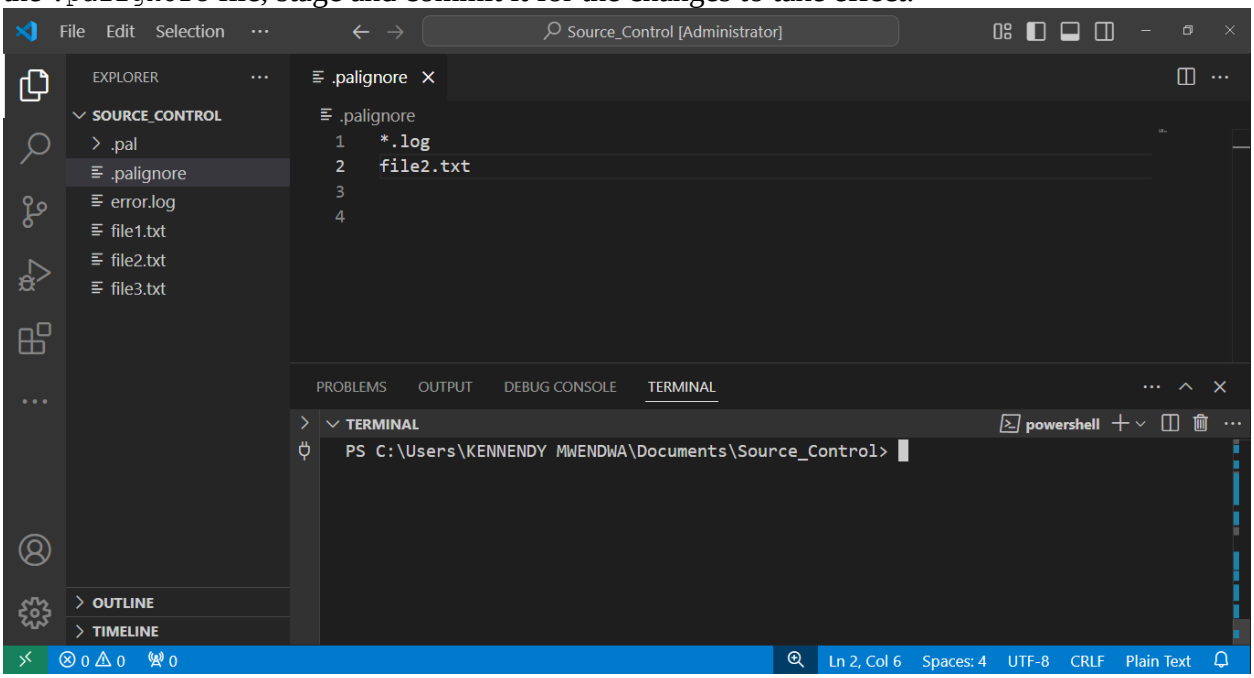
Next, let's merge the branches by running `pal merge <target_branch> <current_branch>`. This command creates a new merge commit and merges the two branches. Ideally, evidence of the merge would appear as a new merge commit in the commit history. However, this functionality is not working currently (more details will be provided in the limitations section).

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal merge master featureOne

  Starting merge operation...
  [INFO] Merging branch 'master' into 'featureOne'...

  [INFO] Resolving merge trees...

  [INFO] Starting tree merge...
  [INFO] Base Tree: 8b9c78c3239cd460639f41b534ceb16c6229d0f2
  [INFO] Current Tree: 5d943e51002b359252cb52180cbe450d9838e6ab
  [INFO] Target Tree: 8b9c78c3239cd460639f41b534ceb16c6229d0f2

  [INFO] Merged tree created successfully with SHA: 5d943e51002b359252cb52180cbe450d9838e6ab

  [INFO] Creating merge commit...

  Merge completed successfully!
  New Commit SHA: 2938d20cf24e88970524827d225c453237eb6b96

  You can view the updated history using `pal log`.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

The next functionality is ignoring files. To demonstrate this, I will create a file specifically to ignore it. The file I created is `error.log`. Additionally, I will ignore `file2.txt` since it is the only text file not currently in the staging area.




To ignore files, you need to create a `.palignore` file at the root of your directory and list the files you want to ignore. In my case, I added all log files (`*.log`) and `file2.txt`. After updating the `.palignore` file, stage and commit it for the changes to take effect.



```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal add .palignore

📂 Repository located successfully.
Adding files to the index...

📄 Preparing files for staging...

✅ Index updated successfully.


✅ Files staged successfully:
  - .palignore

🎉 Use `pal commit` to save your changes.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal commit -m "adding the .palignore file"

[INFO] Generating tree object from the index...
[INFO] Fetching user details...
[INFO] Resolving parent commit...
[INFO] Creating commit object...
Creating a new commit...
```

You can use the `pal check-ignore <name_of_file>` command to verify whether a file is being ignored based on the rules specified in your `.palignore` file.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal check-ignore error.log
Pattern matched: *.log, value: true
error.log: Ignored
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal check-ignore file1.txt
file1.txt: Not Ignored
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal check-ignore file2.txt
Pattern matched: file2.txt, value: true
file2.txt: Ignored
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

You can also try adding the files to the staging area and check if the have been added. They should not be listed in the staging area when you run pal ls-files command.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal add error.log file1.txt file2.txt

📂 Repository located successfully.
Adding files to the index...

📄 Preparing files for staging...

⚠️Removing paths from the index:
  - file1.txt

✅ Index updated successfully.

Pattern matched: *.log, value: true
⚠️Skipped ignored file: error.log
Pattern matched: file2.txt, value: true
⚠️Skipped ignored file: file2.txt

✅ Files staged successfully:
  - file1.txt

🎉 Use `pal commit` to save your changes.
```

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal commit -m "ignoring files"

[INFO] Generating tree object from the index...
[INFO] Fetching user details...
[INFO] Resolving parent commit...
[INFO] Creating commit object...
Creating a new commit...
Commit created successfully with SHA: 407f47e0d7e8f67db85551c1f51b760ab3df2792
[INFO] Updating branch reference...

[SUCCESS] Commit created successfully!
Commit SHA: 407f47e0d7e8f67db85551c1f51b760ab3df2792

🎉 Use `pal log` to view your commit history.

PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal ls-files
[INFO] Found 3 staged file(s):
file3.txt
.palignore
file1.txt
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control>
```

The final functionality is cloning. Running `pal clone "source/directory" "target/directory"` copies the files from the current worktree to the specified target destination directory.

```
TERMINAL                                                    powershell
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> pal clone "C:\Users\KENNENDY MWENDWA\Documents\Source_C
ontrol" "C:\Users\KENNENDY MWENDWA\Documents\Source_Control_Clone"

🔄 Cloning repository...
Source: C:\Users\KENNENDY MWENDWA\Documents\Source_Control
Destination: C:\Users\KENNENDY MWENDWA\Documents\Source_Control_Clone

[INFO] Copying repository metadata...
[SUCCESS] Repository metadata copied successfully.

🔄 Checking out files into the working directory...

✅Checked out: .palignore
✅Checked out: file1.txt
✅Checked out: file3.txt

🎉 Clone completed successfully!

Next steps:
  1. Change to the cloned directory: cd C:\Users\KENNENDY MWENDWA\Documents\Source_Control_Clone
  2. View the repository status: pal status
```

To confirm this, simply navigate to the destination directory and verify the existence of the repository in that location.

```
File  Edit  Selection  ···   Source_Control [Administrator]
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL
TERMINAL                                    powershell - Source_Control_Clone
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control> cd ../
PS C:\Users\KENNENDY MWENDWA\Documents> cd .\Source_Control_Clone\
PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control_Clone> ls


    Directory: C:\Users\KENNENDY MWENDWA\Documents\Source_Control_Clone


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        12/12/2024   6:57 AM               .pal
-a----        12/12/2024   6:57 AM             20 .palignore
-a----        12/12/2024   6:57 AM             64 file1.txt
-a----        12/12/2024   6:57 AM             56 file3.txt


PS C:\Users\KENNENDY MWENDWA\Documents\Source_Control_Clone>
```