Using Git, an IDE, and GitHub for Technical Writers

Frank M. Palinkas, Senior Technical Writer, and Information Developer built and published this document. Use Git from the CMD line, VSCode (or your favorite IDE), and the GitHub Pull Request process to create your work environment and documentation.

Here are the most common Git commands (from the CMD line) and GitHub Actions that Technical Writers use in their work.

Follow these steps:

1. Create a new folder on your main drive (C:\) to house the documentation.
2. Open the CMD line and navigate to this new folder.
3. Go to GitHub and copy the remote repository URL.
4. Open your command line app and run `git clone` - paste the remote repository URL after the command. For example,

```
C:\my-docs\docs>git clone https://github.com/my-repo/my-docs.git
```

5. Hit Enter and wait for the clone to finish.
6. Run `git checkout develop`, `main`, or `master` - ensure you are on the cloned folder branch.
7. Run `git status` - make sure all is in order.
8. Run `git pull` - pull the latest edition of the remote repo branch to your machine.
9. Run `git checkout -b "new branch title goes here"`` - this creates a new branch to contain your work.

Open your IDE

Make sure you will do the work in the new branch created above. Open the folder and files where the work will be done - do the work.

When you are finished creating new or editing current work in your IDE

1. Run `git status` - lists all folders and files where work has been done.
2. Run `git add .` – adds all folders and files where work has been done.
3. Run `git commit -m “message-goes-here“`- to commit the work to Git.
4. Run `git push` - git may show an error, but gives you the correct push instruction below it.
5. Run this new push instruction.
6. When finished, Git gives you a URL to GitHub to create a new Pull Request. Use it in your browser. This pushes your work to GitHub and opens a new Pull Request.

Creating the new Pull Request in GitHub

1. Fill out the necessary information in the Pull Request setup page and create the new PR.
2. Test the rendering of the work using the GitHub Pull Request PREVIEW tool, if available.

3. Wait for GitHub and Gatsby to approve the code and structure accuracy of the work.
4. In GitHub, merge the Pull Request to the internal repo (develop, main, or master).
5. Back in your CMD line window, run `git pull` - pull the merged work to your local machine.
6. In GitHub, merge the Pull Request to the external repo (develop, main, or master repo).