GitHub SOP

# Useful Links

* GitHub Official Reference Guide: <https://git-scm.com/docs>
* TxDOT GitHub Repository: <https://github.com/TxDOT>
* Git install: <https://git-scm.com/download/win>

# Setup Workflow

1. Install Git (see link above)
2. Create GitHub account at <https://github.com>
3. Request Invite to join TxDOT’s GitHub account
4. Open command prompt and 'cd' into the local folder where you want your connection to the repository to reside
   1. cd C:\TxDOT\GIS\Git (This is an example. Your file path may be different)
5. Navigate to the TxDOT/javascript repository and copy the URL using the copy URL button under the colored bar
   1. https://github.com/TxDOT/javascript.git
6. Clone the GitHub repository to your local folder
   1. git clone URL (URL from step 5 above)
7. Create your own dev\_\*user\* branch
   1. git checkout –b dev\_\*user\* (e.g. dev\_tnev)

# Typical Workflow for SPM dev and dev\_user Interactions

This workflow covers scenarios in which you are starting to work in this repository for the first time, or immediately following launch of a new version to ensure you are working on the latest version.

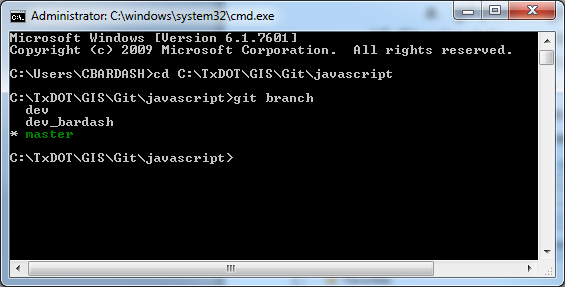
1. Open command prompt
2. cd into local repository
3. Checkout dev to ensure you are starting with the most current version
   1. git checkout dev
4. Pull a copy from GitHub to your local folder
   1. git pull
5. Switch to your dev\_user branch
   1. git checkout dev\_user
6. Merge dev with your dev\_user branch to synch up your branch with dev
   1. git merge dev
7. Make edits
8. Commit your edits to your local
   1. git commit –am “your message here”
9. Push your local copy up to GitHub
   1. git push origin dev\_user
10. Checkout dev so you can merge your branch into the dev branch
    1. git checkout dev
11. Merge your branch with dev
    1. git merge dev\_user
12. Commit to your local
    1. git commit –am “your message here”
13. Push GitHub
    1. git push (enter your username and password)

# Common Actions

**Connect to a Repository**

1. Open the Command prompt
2. CD into the folder where you want your connection to the repository to reside
3. Log into github and copy the URL for the repository
4. Command: git clone *[URL]*

**Check which Branch you are currently working in**

1. Open the Command prompt
2. CD into your local repository
3. Command: git branch

**Add a File to Git**

1. Add the file to the repository folder in windows explorer
2. Open the Command prompt
3. CD into the repository folder
4. Command: git add *[filename]*
5. Alternatively, to add all files not yet added/recognized by Git

Command: git add -A

**Commit Changes** (intentionally save changes locally. Must be done when pulling/merging)

1. Open the Command prompt
2. CD into the repository folder
3. Command: git commit –am “*[text comment about changes made]*”

**Check Status of all Changes**

1. Open the Command prompt
2. CD into the repository folder
3. Command: git status

**Push Changes up to GitHub from Local Machine for Permanent Sharing/Saving**

1. Open the Command prompt
2. CD into the repository folder
3. Command: git push origin *[name of branch being pushed]*
4. Enter your GitHub account username and password as prompted

**Pull Changes from GitHub to your Local Machine** (if you have changes on your local machine they will need to be committed and all conflicts resolved)

1. Open the Command prompt
2. CD into the repository folder
3. Checkout the branch which has the changes you wish to pull with Command: git checkout *[branch name]*
4. Command: git pull

**Create New Branch**

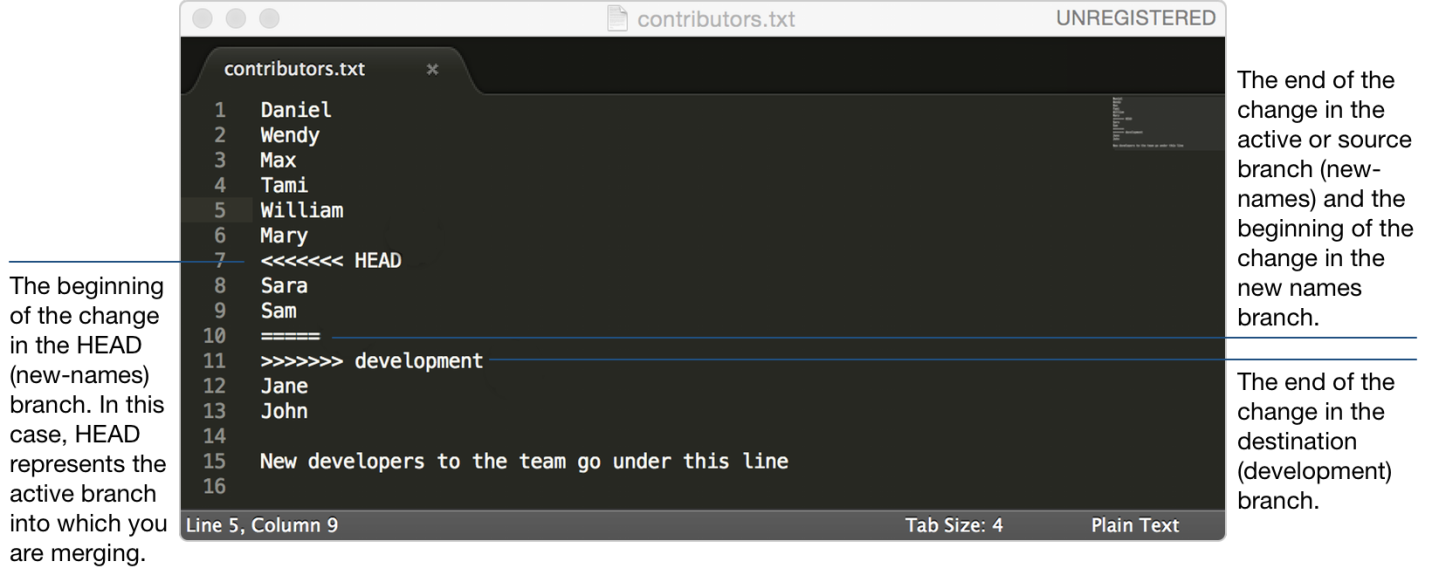
1. Open the Command prompt
2. CD into the repository folder
3. Command: git checkout –b *[name of the new branch]*

**Switch between Branches (Checkout branch)**

1. Open the Command prompt
2. CD into the repository folder (type cd, then space, then the file path to your folder)
3. Command: git checkout *[name of the branch]*

**Merge Branches**

1. Open the Command prompt
2. CD into the repository folder
3. Checkout the branch you want are modifying/merging into (called HEAD)
4. Command: git merge *[name of branch being merged]*
5. If conflicts exist in files, those files will be listed in the Command Prompt. Open those files and resolve the conflicts. They will be identified with:



1. Commit the de-conflicted file with Command: git commit –am “*[text comment about changes made]*”
2. Push the repository back up to GitHub as desired with Command: git push origin *[name of branch being pushed]*

# Definitions

* git push – local to cloud
* git pull – cloud to local
* git merge – merge two branches on your local. Always merge the changes into the branch you currently have check out. For example, to merge your branch with the dev branch, check out dev and merge with your branch.
* git check out – switch to the branch on your local that you want to edit
* git commit – record changes to the repository: a formal save on your local branch. To use Adam’s analogy: “prettying up your chicken scratch.”
* git clone – create a new local repository from and existing GitHub repository.
* git add – add a file that is not yet recognized by git