**Recommended Folder Prep:**

*This has nothing to do with Git, more with staying organized and improving your SSA dev experience.*

If you don't already have one, create a **C:\WS\repos**

We're going to use **C:\WS\repos** as a location for your local git repositories

**C:\WS\repos\demo\_demo>git init**

Initialized empty Git repository in C:/WS/repos/demo\_demo/.git/

**C:\WS\repos\demo\_demo>git status**

On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)

**\*\*\*\*\*\*\*\*\*\*\*After Adding file\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C:\WS\repos\demo\_demo>git status**

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

Mytest.txt

nothing added to commit but untracked files present (use "git add" to track)

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Staging file in Git\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C:\WS\repos\demo\_demo>git add .**

C:\WS\repos\demo\_demo>git commit -m "first file added"

[master (root-commit) dc03341] first file added

1 file changed, 1 insertion(+)

create mode 100644 Mytest.txt

**\*\*\*\*\*\*\*Check Status Again\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C:\WS\repos\demo\_demo>git status**

On branch master

nothing to commit, working tree clean

C:\WS\repos\demo\_demo>git init

Reinitialized existing Git repository in C:/WS/repos/demo\_demo/.git/

**\*\*\*\*\*\*\*\*\*Existing Tracked Git Files\*\*\*Push to master on Bitbucket\*\*\*\*\*\*\*\*\*\*\***

**C:\WS\repos\demo\_demo>git remote set-url origin ssh://git@stash.ba.ssa.gov/~4064**

**97/demo\_demo.git**

**C:\WS\repos\demo\_demo>git push -u origin master**

Counting objects: 3, done.

Writing objects: 100% (3/3), 241 bytes | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To ssh://stash.ba.ssa.gov/~406497/demo\_demo.git

\* [new branch] master -> master

Branch master set up to track remote branch master from origin.

**C:\WS\repos\demo\_demo>git status**

On branch master

Your branch is up-to-date with 'origin/master'.

nothing to commit, working tree clean

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*Creating New Branch\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C:\WS\repos\demo\_demo>git branch Mydemo**

**C:\WS\repos\demo\_demo>git status**

On branch master

Your branch is up-to-date with 'origin/master'.

nothing to commit, working tree clean

**\*\*\*\*\*\*\*\*Checkingout in New Branch\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C:\WS\repos\demo\_demo>git checkout mydemo**

Switched to branch 'mydemo'

C:\WS\repos\demo\_demo>git status

On branch mydemo

nothing to commit, working tree clean

C:\WS\repos\demo\_demo>git status

On branch mydemo

nothing to commit, working tree clean

**\*\*\*\*\*\*\*\*\*\*\*\*After modifying File\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C:\WS\repos\demo\_demo>git status**

On branch mydemo

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: Mytest.txt

no changes added to commit (use "git add" and/or "git commit -a")

**\*\*\*\*\*\*\*\*\*\*Staging Modified file in my new Branch\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

C:\WS\repos\demo\_demo>**git commit -a -m "Second Change"**

[mydemo f793e7d] Second Change

1 file changed, 2 insertions(+), 1 deletion(-)

**C:\WS\repos\demo\_demo>git status**

On branch mydemo

nothing to commit, working tree clean

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*Pushing new Branch to Bitbucket\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C:\WS\repos\demo\_demo>git push -u origin mydemo**

Counting objects: 3, done.

Writing objects: 100% (3/3), 281 bytes | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

remote:

remote: Create pull request for mydemo:

remote: https://stash.ba.ssa.gov/users/406497/repos/demo\_demo/compare/commits?

sourceBranch=refs/heads/mydemo

remote:

To ssh://stash.ba.ssa.gov/~406497/demo\_demo.git

\* [new branch] mydemo -> mydemo

Branch mydemo set up to track remote branch mydemo from origin.

C:\WS\repos\demo\_demo>

**GIT common commands:**

**start a working area (as per SSA Standard “C:ws\repos\ProjectName”)**

**clone Clone a repository into a new directory**

**init Create an empty Git repository or reinitialize an existing one**

**work on the current change (see also: git help everyday)**

**add Add file contents to the index**

**mv Move or rename a file, a directory, or a symlink**

**reset Reset current HEAD to the specified state**

**rm Remove files from the working tree and from the index**

**Examine the history and state (see also: git help revisions)**

**bisect Use binary search to find the commit that introduced a bug**

**grep Print lines matching a pattern**

**log Show commit logs**

**show Show various types of objects**

**status Show the working tree status**

**Grow, mark and tweak your common history**

**branch List, create, or delete branches**

**checkout Switch branches or restore working tree files**

**commit Record changes to the repository**

**diff Show changes between commits, commit and working tree, etc**

**merge Join two or more development histories together**

**rebase Reapply commits on top of another base tip**

**tag Create, list, delete or verify a tag object signed with GPG**

**Collaborate (see also: git help workflows)**

**fetch Download objects and refs from another repository**

**pull Fetch from and integrate with another repository or a local branch**

**push Update remote refs along with associated objects**

**Git log –oneline –decorate –all**

**Shows all logs of master and branch with git HEAD location**

**git aliases used git commands:**

co **=** checkout # Checkout a branch

cob **=** checkout -b # Checkout a new not yet existing branch

f **=** fetch -p # Fetch from a repository and prune any remote-tracking references that no longer exist on the remote.

c **=** commit # Commit you changes

p **=** push # Push you changes to a remote

ba **=** branch -a # List both remote-tracking branches and local branches.

bd **=** branch -d # Delete a branch only if it has been merged

bD **=** branch -D # Force delete a branch

dc **=** diff --cached # Display the staged changes.

st = status –sb # Checking the status of your git commit

git add -p # Stage your git changes in patches