

Git Update + Rebase Workflow

Goal: Update your branch with the latest Development changes and push safely.

A) If you have uncommitted work

1) Save your work

- Option 1 (commit):

```
git add .
```

```
git commit -m "WIP"
```

- Option 2 (stash):

```
git stash -u
```

B) Update local Development

2) Switch to Development
git checkout Development

3) Pull latest from GitHub

```
git pull --rebase origin Development
```

C) Rebase your feature branch onto Development

4) Switch back to your branch
git checkout <your-branch>

5) Rebase onto Development

```
git rebase Development
```

D) If conflicts happen

6) Fix conflicts in files

7) Stage fixed files

```
git add .
```

8) Continue rebase

```
git rebase --continue
```

If you want to abort rebase:

```
git rebase --abort
```

E) Push your branch

9) Push to GitHub

```
git push -u origin <your-branch>
```

If you already pushed before (history changed by rebase):

```
git push --force-with-lease
```

Notes

- Rebase keeps history clean. Merge is fine if your team prefers it.

- Always use --force-with-lease (not --force) after rebasing.

- If you used stash, re-apply it after rebase:

```
git stash apply
```

Quick Merge Alternative (no rebase)

- git checkout Development

- git pull origin Development

- git checkout <your-branch>

- git merge Development

- resolve conflicts, git add ., git commit

- git push