Git Push & Pull Scripts Guide

This guide explains how to use the `push.sh` and `pull.sh` scripts to automate your Git workflow.

Prerequisites

- Git Bash, WSL, or any Bash-compatible terminal on Windows
- Make scripts executable: `chmod +x push.sh pull.sh`

Quick Reference

Command	Action
`./push.sh`	Push to **current branch**
`./push.sh -b <branch>`	Push to **specific branch**
`./pull.sh`	Pull from **current branch**
`./pull.sh -b <branch>`	Pull from **specific branch**

Push Script (`push.sh`)

Basic Usage

```
# Push to current branch with auto-generated commit message
./push.sh

# Push to current branch with custom commit message
./push.sh -m "Your commit message"
./push.sh "Your commit message"  # Also works
```

Push to Specific Branch

```
# Push to a specific branch (e.g., feature-branch)
./push.sh --branch feature-branch
./push.sh -b feature-branch

# Push to specific branch with commit message
./push.sh -b feature-branch -m "Added new feature"
```

Push to Master

```
# If you're already on master
./push.sh -m "Update master branch"

# If you're on another branch but want to push to master
./push.sh --branch master -m "Merge changes to master"
```

Options

Option	Short	Description
`--branch`	`-b`	Target branch to push to
`--message`	`-m`	Commit message
`--help`	`-h`	Show help message

What Happens When You Push

1. **Without** `--branch` : Pushes to your **current branch**
2. **With** `--branch <name>` :
   - If branch exists locally → switches to it
   - If branch exists only on remote → creates local tracking branch
   - If branch doesn't exist → creates new branch
   - Your uncommitted changes are **automatically stashed** and restored

---

Pull Script (`pull.sh`)

Basic Usage

```
# Pull latest changes for current branch
./pull.sh
```

Pull from Specific Branch

```
# Pull from a specific branch
./pull.sh --branch feature-branch
./pull.sh -b feature-branch
```

Pull from Master

```
# If you're already on master
./pull.sh

# If you're on another branch but want to pull master updates
./pull.sh --branch master
```

Options

Option	Short	Description
`--branch`	`-b`	Target branch to pull from
`--help`	`-h`	Show help message

What Happens When You Pull

1. **Without** `--branch` : Pulls from your **current branch**
2. **With** `--branch <name>` :
   - Switches to the specified branch
   - Pulls latest changes from that branch
   - Your uncommitted changes are **automatically stashed** and restored

---

Branch Behavior Summary

When Does It Use Master?

Scenario	Branch Used
You're on `master` and run `./push.sh`	`master`
You're on `master` and run `./pull.sh`	`master`
You run `./push.sh -b master`	`master`
You run `./pull.sh -b master`	`master`

When Does It Use Another Branch?

Scenario	Branch Used
You're on `feature-x` and run `./push.sh`	`feature-x`
You're on `feature-x` and run `./pull.sh`	`feature-x`
You run `./push.sh -b develop`	`develop`
You run `./pull.sh -b develop`	`develop`

Safety Features

Both scripts include safety measures to protect your work:

1. **Auto-stashing**: Uncommitted changes are automatically stashed before branch switching
2. **Auto-restore**: Stashed changes are restored after the operation completes
3. **Error handling**: If something fails, stashed changes are restored
4. **No force push**: Scripts never use `--force`, keeping remote history safe
5. **Branch isolation**: Operations only affect the target branch

---

Common Workflows

Feature Branch Workflow

```
# 1. Pull latest master
./pull.sh -b master

# 2. Create and switch to feature branch (manually)
git checkout -b feature-new-login

# 3. Make your changes...

# 4. Push feature branch
./push.sh -m "Implement new login feature"

# 5. Later, pull updates from master while on feature branch
./pull.sh -b master
git checkout feature-new-login
git merge master
```

Quick Update Master

```
# Pull and push master quickly
./pull.sh -b master
./push.sh -b master -m "Quick fix"
```

Sync Multiple Branches

```
# Update master
./pull.sh -b master

# Update develop
./pull.sh -b develop
```

```
# Push your current branch
./push.sh -m "My changes"
```

Troubleshooting

"Push failed. You may need to pull first"

```
./pull.sh  # Pull first
./push.sh -m "Your message"  # Then push
```

"Could not restore stashed changes"

```
git stash list  # See stashed changes
git stash pop   # Manually restore
```

"Branch does not exist"

```
# For pull: The branch must exist on remote
git fetch origin
git branch -a  # List all branches

# For push: A new branch will be created automatically
```

Examples

```
# Example 1: Daily workflow on master
./pull.sh                    # Get latest
# ... make changes ...
./push.sh -m "Daily update"  # Push changes

# Example 2: Work on feature branch
./push.sh -b feature-auth -m "Add authentication"

# Example 3: Quick sync with master while on feature branch
./pull.sh -b master          # Updates master locally
```

```
git checkout feature-auth    # Go back to your branch
git merge master             # Merge master into your branch
```