Contents

Git Started with GitHub

What is GIT/Core Concepts

What is Git?

It's a distributed source code control system.

It does not rely on a central server - although many people usually have one.

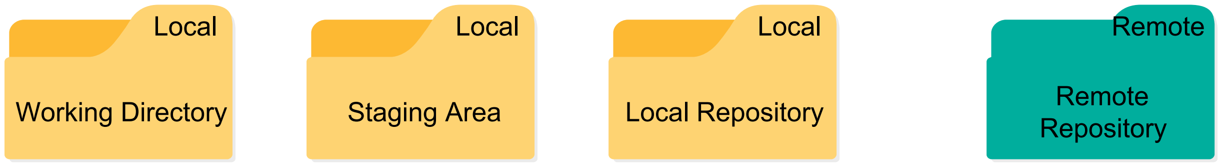
The developer can work happily by themselves, checking in/out/merging/undoing to their heart's content without ever affecting the central repository or other developers.

When they have finished - they 'push' their changes to the central server for others to download.

There are only a few network commands, most are local and do not need a network connection.

It's very fast, free and open source.

Core Concepts



This is the version of the repository which you are working on

This is sort of a pre-commit area - you can use to place changes ready for commit

This is your changes commited - ready for making available to other developers

This is your changes commited - ready for making available to other developers

The above is a conceptual view of the different parts of GIT.

'Working Directory' is what you actually change/compile/run.

Staging area is somewhere you can put your changes when you have finished - but not ready to make available to other developers.

Local Repository is a complete copy of every change ever made. You commit your changes to the local repository - then you ready - you can push the changes over the network to make them available to other developers. When other developers push their changes - your client tells you there are changes to 'pull'.

Remote Repository - is the repository available to other users.

Installation/Initialisation

Installation

Installing is very simple - simply navigate to http://gitforwindows.org, install - there are options during the install - these are fairly self explanatory.

When finished - open up 'git bash shell' and type

```
> git version
```

This should show you the installed version.

Hosting Servers

You can host your own if you want.

There are lots of free and paid servers out there.

The most common is github, the free licence is ok but forces you to make all your source made public. If you want them to be private - it costs - but only about £7 / month.

Take a look at www.github.com

Initialising GIT and Project Directory

I tend to put all projects in a 'projects' sub-directory of my home area. Start git bash shell.

```
> mkdir projects
```

Now you need to configure your name and email (same as in git hub if you've done it that way).

```
> git config --global user.name "Des Cooke"
> git config -- global user.email "des.cooke@kcom.com"
```

to confirm

```
> git config --global --list
```

Creating Repositories

Creating New Repository in GitHub

You can use - github - create an account - then login - click create repository.

Once created - you can take the repository url (bottom left)

Then in git bash shell type

```
> git clone <paste url>
> cd <project name>
> git status
```

Creating New Repository from Git Bash Shell

Start git bash shell

```
> cd projects
> git init myproject
> cd myproject
> git status
> git push origin master
```

This will create a local repository and file system (obviously no working files yet)

Adding an existing project to GIT

Start git bash shell

```
> cd projects
> cd myproject
> git init
> git add .
> git commit -m "initial"
> git push origin master
```

This will create a local repository and file system (obviously no working files yet)

Removing a project from GIT

Simply remove the .git sub-directory (removes local repository) and remove from github (removes remote repository)

Adding new files

Adding new files

If you have just created a file in your working directory - type

```
> git status <filename>
```

This tells you the file is 'untracked' (not managed via git)

```
> git add <filename>
> git status <filename>
```

Tells you now it is in the staging area - we can now commit it to the local repository

```
> git commit -m "initial" <filename>
> git status <filename>
```

tells you now the working directory is clean and no outstanding changes - but it does say that the local repository is 1 commit ahead of the remote repository - so we push it using

```
> git push origin master <filename>
```

This pushes the file to the origin branch on the remote server from the master server on local repository.

Since we are doing a network - GIT may ask you for your github password