

What are Git refs and refspec?

Indo Yoon

2023-01-18

Git manages branches and commits of the repo in the local filesystem.

How does it stores and references them?

Why do you care?

Jenkins Gerrit plugin! 🥰

1. refs

Easy to find commit history!

```
$ git log
```

```
commit fc593648dc05cecb3a2fa425ef68c826a440a9ad  
Author: Indosaram <freedomzero91@gmail.com>  
Date: Sun Oct 2 15:15:12 2022 +0900
```

What about a certain branch's history?

```
$ git rev-parse main  
fc593648dc05cecb3a2fa425ef68c826a440a9ad
```

3 types of refs

refs : a feature for easily referencing the commit

- heads : local branch
- remotes : remote branch
- tags : "usually" for releases, often used as **v1.0**

branch : dynamic, **tag** : static

They are stored in each folder under `.git/refs`.

```
.git/refs
├── heads
│   └── main
├── remotes
│   └── origin
│       ├── HEAD
│       └── main
└── tags
```

If you want to find a `HEAD` of the local branch:

```
$ cat heads/main  
fc593648dc05cecb3a2fa425ef68c826a440a9ad
```

It's the same hash from `git log` !

Local-only refs

- `HEAD` – Currently checked-out commit / branch
- `FETCH_HEAD` – Latest fetched branch from the remote
- `ORIG_HEAD` – A backup reference of `HEAD`
- `MERGE_HEAD` – The source branch when `git merge` is used
- `CHERRY_PICK_HEAD` – For cherry-picking

HEAD pointing the branch and the commit that **git** is currently referencing.

```
$ cat .git/HEAD
ref: refs/heads/master

$ git checkout remote
Switched to branch 'remote'

$ cat .git/HEAD
ref: refs/heads/remote
```

refname : e.g. **master**, **heads/master**, **refs/heads/master**

Relative refs

```
git show HEAD~2
```

2. refsPEC

 between the local and the remote

When a Github repo is being initialized:

...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Indosaram/test.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/Indosaram/test.git
git branch -M main
git push -u origin main
```



"Hi Git, please consider **origin** as a remote branch of my repo."

```
git remote add origin https://github.com/Indosaram/test.git
```

"Please reflect my local change to the remote."

```
git push -u origin main
```

git's refspec

git's default refspec form: `[+]<src>:<dest>`

- `<src>` : local branch
- `<dest>` : remote branch

Optional `+` prevents the remote branch from fast-forward update

```
<src>:<dest>
```

An example

```
$ cat .git/config  
  
...  
  
[remote "origin"]  
    url = git@github.com:Indosaram/test.git  
    fetch = +refs/heads/*:refs/remotes/origin/*  
  
...
```

- `+` : Do not enforce fast-forward
- `*` : Update all local branches using all remote branches. In a repo with many branches, `git fetch` updates all of them because of this.

Gerrit's refsPEC

Gerrit uses change and patchset. A local commit needs information to which change it should be appended.

To create a new change or append a patchset:

```
git push origin HEAD:refs/for/master
```

If a change is created, `change-id` is generated from the Gerrit.

```
$ git log -1
commit 29a6bb1a059aef021ac39d342499191278518d1d
Author: A. U. Thor <author@example.com>
Date: Thu Aug 20 12:46:50 2009 -0700

    Improve foo widget by attaching a bar.

Change-Id: Ic8aaa0728a43936cd4c6e1ed590e01ba8f0fbf5b
```

If `commit-msg` hook is installed,
local git client will generate the change-id.

Gerrit events

Gerrit does not have fancy CI/CD integration. Instead, there is hooks.
The hooks notify events from Gerrit.

[List of hooks](#)

ref-update

`ref-update` hook is invoked when Gerrit receives the following 3 events.

- Direct push to the branch
- Non fast-forward update
- Deletion of `ref`

ref-update

Triggered whenever a commit is pushed.

commit-received

If the hook exits with non-zero code, the remote is not updated. Therefore, not every commit push is updated in the remote branch.

Reference

- [Refs and the Reflog](#)
- [10.5 Git Internals - The Refspec](#)
- [Supported Hooks](#)