

## Sharing Tags

By default, the `git push` command doesn't transfer tags to remote servers. You will have to explicitly push tags to a shared server after you have created them. This process is just like sharing remote branches — you can run `git push origin <tagname>`.

```
$ git push origin v1.5
```

```
Counting objects: 14, done.
```

```
Delta compression using up to 8 threads.
```

```
Compressing objects: 100% (12/12), done.
```

```
Writing objects: 100% (14/14), 2.05 KiB | 0 bytes/s, done.
```

```
Total 14 (delta 3), reused 0 (delta 0)
```

```
To git@github.com:schacon/simplegit.git
```

```
* [new tag]      v1.5 -> v1.5
```

If you have a lot of tags that you want to push up at once, you can also use the `--tags` option to the `git push` command. This will transfer all of your tags to the remote server that are not already there.

```
$ git push origin --tags
```

```
Counting objects: 1, done.
```

```
Writing objects: 100% (1/1), 160 bytes | 0 bytes/s, done.
```

```
Total 1 (delta 0), reused 0 (delta 0)
```

```
To git@github.com:schacon/simplegit.git
```

```
* [new tag]      v1.4 -> v1.4
```

```
* [new tag]      v1.4-lw -> v1.4-lw
```

Now, when someone else clones or pulls from your repository, they will get all your tags as well.

<b>Note</b>	<b>git push pushes both types of tags</b>  <b>git push &lt;remote&gt; --tags will push both lightweight and annotated tags. There is currently no option to push only lightweight tags, but if you use git push &lt;remote&gt; --follow-tags only annotated tags will be pushed to the remote.</b>
-------------	--

## Deleting Tags

To delete a tag on your local repository, you can use `git tag -d <tagname>`. For example, we could remove our lightweight tag above as follows:

```
$ git tag -d v1.4-lw
```

Deleted tag 'v1.4-lw' (was e7d5add)

Note that this does not remove the tag from any remote servers. There are two common variations for deleting a tag from a remote server.

The first variation is `git push <remote> :refs/tags/<tagname>`:

```
$ git push origin :refs/tags/v1.4-lw
```

```
To /git@github.com:schacon/simplegit.git
```

```
- [deleted]    v1.4-lw
```

The way to interpret the above is to read it as the null value before the colon is being pushed to the remote tag name, effectively deleting it.

The second (and more intuitive) way to delete a remote tag is with:

```
$ git push origin --delete <tagname>
```

## Checking out Tags

If you want to view the versions of files a tag is pointing to, you can do a git checkout of that tag, although this puts your repository in “detached HEAD” state, which has some ill side effects:

```
$ git checkout v2.0.0
```

Note: switching to 'v2.0.0'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable `advice.detachedHead` to `false`

HEAD is now at 99ada87... Merge pull request #89 from schacon/appendix-final

```
$ git checkout v2.0-beta-0.1
```

Previous HEAD position was 99ada87... Merge pull request #89 from schacon/appendix-final

HEAD is now at df3f601... Add atlas.json and cover image

In “detached HEAD” state, if you make changes and then create a commit, the tag will stay the same, but your new commit won’t belong to any branch and will be unreachable, except by the exact commit hash. Thus, if you need to make changes — say you’re fixing a bug on an older version, for instance — you will generally want to create a branch:

```
$ git checkout -b version2 v2.0.0
```

Switched to a new branch 'version2'

If you do this and make a commit, your `version2` branch will be slightly different than your `v2.0.0` tag since it will move forward with your new changes, so do be careful.