

Tidying Repositories

 coursera.org/learn/git-distributed-development/supplement/ztXmT/tidying-repositories

As your project grows through a series of commits, your repository may grow large in size. You can optimize and compact your repository by issuing **git gc**, as in:

```
$ du -shc .git
```

```
47M    .
```

```
47M    total
```

```
$ git gc
```

```
Counting objects: 8, done.
```

```
Compressing objects: 100% (8/8), done.
```

```
Writing objects: 100% (8/8), done.
```

```
Total 8 (delta 2), reused 0 (delta 0)
```

```
$ du -shc .git
```

```
29M    .git
```

```
29M    total
```

where **gc** stands for garbage collection.

You can also check your repository for certain kinds of errors with the command **git fsck**. The most likely and harmless errors it will find will be dangling objects; while these are sometimes useful for recovering corrupted repositories, they can generally be safely removed with **git prune** as in:

```
$ git prune -n
```

```
$ git prune
```

where the first command just checks to see what would be done, and if you are happy with it, you can issue the second pruning command.

