

**NAME**

git-stripspace - Remove unnecessary whitespace

**SYNOPSIS**

*git stripspace* [-s | --strip-comments]

*git stripspace* [-c | --comment-lines]

**DESCRIPTION**

Read text, such as commit messages, notes, tags and branch descriptions, from the standard input and clean it in the manner used by Git.

With no arguments, this will:

⊕

trailing whitespace from all lines

⊕

multiple consecutive empty lines into one empty line

⊕

empty lines from the beginning and end of the input

⊕

a missing `\n` to the last line if necessary.

In the case where the input consists entirely of whitespace characters, no output will be produced.

**NOTE:** This is intended for cleaning metadata, prefer the **--whitespace=fix** mode of **git-apply(1)** for correcting whitespace of patches or files in the repository.

**OPTIONS**

-s, --strip-comments

Skip and remove all lines starting with comment character (default #).

-c, --comment-lines

Prepend comment character and blank to each line. Lines will automatically be terminated with a newline. On empty lines, only the comment character will be prepended.

**EXAMPLES**

Given the following noisy input with `$` indicating the end of a line:

```
|A brief introduction $
| $
|$
|A new paragraph$
|# with a commented-out line $
|explaining lots of stuff.$
|$
|# An old paragraph, also commented-out. $
| $
|The end.$
| $
```

Use `git stripSpace` with no arguments to obtain:

```
|A brief introduction$
|$
|A new paragraph$
|# with a commented-out line$
|explaining lots of stuff.$
|$
|# An old paragraph, also commented-out.$
|$
|The end.$
```

Use `git stripSpace --strip-comments` to obtain:

```
|A brief introduction$
|$
|A new paragraph$
|explaining lots of stuff.$
|$
|The end.$
```

## **GIT**

Part of the `git(1)` suite