

sed (search and replace text)

This command allows you to search and replace strings in text files. Here is an example of its usage:

```
sed `s/bad/good/g` file.txt
```

This will output file.txt with every occurrence of the string *bad* replaced by *good*. The strings are interpreted as regular expressions (see Appendix A in the [printed book](#)). The *s* in the above invocation refers to *sed*'s search and replace command. *sed* accepts many other commands, but *s* is the one that's most commonly used.

By default, *s* will replace only the first string on each input line. In most cases that's not what you want. The above example contains the modifier *g*("global"), which widens the scope of the *s* command to the whole of every input line.

sed does not alter input files by default. In the above example, file.txt is not modified, as the search/replace operation only affects the output that you see on the terminal. You can redirect *sed*'s output to another file to save the changes, for example:

```
sed `s/bad/good/g` file.txt > new.txt
```

You can also instruct *sed* to edit the input file directly, using the option *-i*("in place"):

```
sed -i.bak `s/bad/good/g` file.txt
```

This will save the result of the search/replace operation in file.txt, overwriting the file's previous contents. A backup of the original file ending in .bak will be created. If you don't supply a suffix such as .bak to the *-i* option, *sed* won't create a backup. You should always specify a suffix to avoid butchering files by mistake.