

# Setup folder rights

For security reasons, there is no need for web server to have access to write to other directories.

## Set the owner and clean directories

First, change `www-data` to your web server user.

## Clean the cache/ and logs/ directories

```
$ rm -rf app/cache/* app/logs/*
```

## Use the right option according to your system.

### A. Using ACL on a *Linux/BSD* system that supports `chmod +a`

#### Using ACL on a Linux/BSD system that supports `chmod +a`

```
$ sudo chmod +a "www-data allow
delete,write,append,file_inherit,directory_inherit" \
  app/cache app/logs web
$ sudo chmod +a "`whoami` allow
delete,write,append,file_inherit,directory_inherit" \
  app/cache app/logs web
```

### B. Using ACL on a *Linux/BSD* system that does not support `chmod +a`

Some systems don't support `chmod +a`, but do support another utility called `setfacl`. You may need to enable ACL support on your partition and install `setfacl` before using it (as is the case with Ubuntu), in this way:

#### Using ACL on a Linux/BSD system that does not support `chmod +a`

```
$ sudo setfacl -R -m u:www-data:rwX -m u:`whoami`:rwX \
  app/cache app/logs web
$ sudo setfacl -dR -m u:www-data:rwX -m u:`whoami`:rwX \
  app/cache app/logs web
```

### C. Using `chown` on *Linux/BSD/OS X* systems that don't support ACL

Some systems don't support ACL at all. You will need to set your web server's user as the owner of the required directories:

- Set the owner and clean directories
  - Clean the cache/ and logs/ directories
- Use the right option according to your system.
  - A. Using ACL on a Linux/BSD system that supports `chmod +a`
  - B. Using ACL on a Linux/BSD system that does not support `chmod +a`
  - C. Using `chown` on Linux/BSD/OS X systems that don't support ACL
  - D. Using `chmod` on a Linux/BSD/OS X system where you can't change owner
  - E. Setup folder rights on Windows

### Using chown on Linux/BSD/OS X systems that don't support ACL

```
$ sudo chown -R www-data:www-data app/cache app/logs web
$ sudo find {app/{cache,logs},web} -type d | xargs sudo
chmod -R 775
$ sudo find {app/{cache,logs},web} -type f | xargs sudo
chmod -R 664
```

## D. Using chmod on a *Linux/BSD/OS X* system where you can't change owner

If you can't use ACL and aren't allowed to change owner, you can use chmod, making the files writable by everybody. Note that this method really isn't recommended as it allows any user to do anything:

### Using chmod on a Linux/BSD/OS X system where you can't change owner

```
$ sudo find {app/{cache,logs},web} -type d | xargs sudo
chmod -R 777
$ sudo find {app/{cache,logs},web} -type f | xargs sudo
chmod -R 666
```

When using chmod, note that newly created files (such as cache) owned by the web server's user may have different/restrictive permissions. In this case, it may be required to change the umask so that the cache and log directories will be group-writable or world-writable (umask(0002) or umask(0000) respectively).

It may also be possible to add the group ownership inheritance flag so new files inherit the current group, and use 775/664 in the command lines above instead of world-writable:

### It may also be possible to add the group ownership inheritance flag

```
$ sudo chmod g+s {app/{cache,logs},web}
```

## E. Setup folder rights on Windows

For your choice of web server you'll need to make sure web server user has read access to `<root-dir>`, and write access to the following directories:

- app/cache
- app/logs