Files

Read, write atomically and delete a file with File

Anonymous · 05/11/2026

Files

```
use Derafu\File\File;
```

Method	What it does
`File::read($path)`	The whole content of a file.
`File::write($path, $content, $permissions = null)`	Writes a file atomically.
`File::delete($path)`	Deletes a file or a symbolic link.

Reading

```
$content = File::read('/var/app/data/config.json');
```

A symbolic link to a file is followed. The failures are:

Case	Exception
The file does not exist (a broken link too).	`FileNotFoundException`
The path is a directory, or the file can not be read.	`FileAccessException`
The path has a null character.	`PathException`
The reading fails.	`FileException`

An empty file is the empty string.

Writing

```
File::write('/var/app/data/config.json', $json);
File::write('/var/app/data/secret.key', $key, 0600);
```

The content goes to a temporary file in the same directory, and then it **replaces** the target with a rename. Another process sees the old file or the new one, never a half-written file, and if the process

is interrupted the old file is still there. If anything fails the temporary file is deleted and the original stays as it was.

- The directories that are missing are created (see `Directory::ensure()`).
- The permissions are `0666` minus the `umask` of the process, unless they are given.
- A directory that can not be written is a `FileAccessException`; one that can not be created, or a move that fails (a file can not replace a directory), is a `FileException`.
- A path with a null character is a `PathException`.

Deleting

```
File::delete('/tmp/export.csv');
```

A path that does not exist is already deleted: nothing happens. A symbolic link is deleted and what it points to stays. A directory is a `FileAccessException` (see `Directory::remove()`); a deletion that fails is a `FileException`.

Last updated on 05/11/2026