

# Package ‘ConfigParser’

May 7, 2026

**Title** Package to Parse an INI File, Including Variable Interpolation

**Version** 1.0.0

**Description** Enhances the 'ini' package by adding the ability to interpolate variables. The INI configuration file is read into an R6 ConfigParser object (loosely inspired by Python's ConfigParser module) and the keys can be read, where '%(...)' instances are interpolated by other included options or outside variables.

**Depends** R (>= 3.0.0), ini, R6

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Suggests** testthat

**RoxygenNote** 6.0.1.9000

**URL** <https://github.com/hhoeflin/ConfigParser>

**BugReports** <https://github.com/hhoeflin/ConfigParser/issues>

**NeedsCompilation** no

**Author** Holger Hoefling [aut, cre]

**Maintainer** Holger Hoefling <hhoeflin@gmail.com>

**Repository** CRAN

**Date/Publication** 2017-08-22 14:02:31 UTC

## Contents

configParser-class . . . . .	2
<b>Index</b>	<b>4</b>

---

configParser-class     *R6 object to parse an INI file*

---

## Description

This class creates an object that represents an INI file. It also supports interpolation of variables of the form `%(...)`s. More detail on how the interpolation works is given in the explanation of method `get`.

## Methods

`new(init = NULL, optionxform = tolower)` Initializes the object

Initializes the object, optionally with pre-set variables given as a list or an environment

### Parameters:

**init** A named character vector, named list or an environment of variables to pre-set (the will be put into the DEFAULT section)

**optionxform** Function through which all option and section names are funneled. By default `tolower` to make all names lowercase. To switch off, pass in `identity`

`read(filepath, encoding = getOption("encoding"))` Read in an INI file

Reads the INI given into the object. It will be added to the internal storage of the object. If applying these functions several times, later definitions overwrite those that were read earlier.

### Parameters:

**filepath** The path to the INI file.

**encoding** The encoding to use

### Return:

The object, but with the added data of the file

`write(filepath, encoding = getOption("encoding"))` Write an INI file

Write the ConfigParser object into the INI file. It writes all the different sections using uninterpolated variables.

### Parameters:

**filepath** The path to the INI file.

**encoding** The encoding to use

### Return:

The ConfigParser object itself

`get(option, fallback, section = "default", interpolate = TRUE)` Get the value of a option

### Parameters:

**option** The option for which to get the value

**fallback** The fallback value to return if there is no value for the option. If missing, an error will be thrown if the option has no value.

**section** The section (or several) from which to read the option. It will try to read the option in the given section from first to last, with DEFAULT always being the last

**interpolate** Should the values be interpolated. This will try to replace variables of the form `%(...)s`

**Return:**

The value of the option

`getboolean(option, fallback, section = "default", interpolate = TRUE)` Same as `get`, but results coerced to a logical.

`getfloat(option, fallback, section = "default", interpolate = TRUE)` Same as `get`, but the result coerced to a float.

`set(option, value, section, error_on_new_section = TRUE)` Set option to a value in a section

Sets an option to the given value (which can include variables for interpolation) in the given section.

**Parameters:**

**option** Name of the option to set (can be a character vector)

**value** Value of the option (same length as option)

**section** Character vector of length 1

**error\_on\_new\_section** Should an error be raised if the section does not exist yet

**Return:**

Return of the adjusted `ConfigParser` object itself

# Index

ConfigParser (configParser-class), [2](#)  
configParser-class, [2](#)