

**NAME**

Config::File - Parse a simple configuration file

**SYNOPSIS**

```
use Config::File;

my $config_hash = Config::File::read_config_file($configuration_file);
```

**DESCRIPTION**

`read_config_file` parses a simple configuration file and stores its values in an anonymous hash reference. The syntax of the configuration file is as follows:

```
# This is a comment
VALUE_ONE = foo
VALUE_TWO = $VALUE_ONE/bar
VALUE_THREE = The value contains a \# (hash). # This is a comment.
```

Options can be clustered when creating groups:

```
CLUSTER_ONE[data] = data cluster one
CLUSTER_ONE[value] = value cluster one
CLUSTER_TWO[data] = data cluster two
CLUSTER_TWO[value] = value cluster two
```

Then values can be fetched using this syntax:

```
$hash_config->{CLUSTER_ONE}{data};
```

There can be as many sub-options in a cluster as needed.

```
BIG_CLUSTER[part1][part2][part3] = data
```

is fetched by: `$hash_config->{BIG_CLUSTER}{part1}{part2}{part3};`

There are a couple of restrictions as for the names of the keys. First of all, all the characters should be alphabetic, numeric, underscores or hyphens, with square brackets allowed for the clustering. That is, the keys should conform to `/^[A-Za-z0-9_-]+$/`

This means also that no space is allowed in the key part of the line.

```
CLUSTER_ONE[data] = data cluster one # Right
CLUSTER_ONE[ data ] = data cluster one # Wrong
```

**Function read\_config\_file****Syntax**

```
Config::File::read_config_file($file);
```

**Arguments**

`$file` is the configuration file.

**Return value**

This function returns a hash reference. Each key of the hash is a value defined in the configuration file.

**Description**

`read_config_file` parses a configuration file and sets up some values in a hash reference.

**NOTES****Function not exported by default**

In versions up to 1.0, the function `read_config_file` was exported to the calling program's namespace - Starting in version 1.1, nothing is exported by default. You can either fully qualify `read_config_file` or explicitly import it into your namespace:

```
Fully qualifying read_config_file
use Config::File;
```

```
my $config_hash = Config::File::read_config_file($configuration_file);  
Explicitly importing read_config_file  
use Config::File qw(read_config_file);  
  
my $config_hash = read_config_file($configuration_file);
```

### **Migrated away from ConfigFile into Config::File**

As of version 1.4, in order to include this module in the CPAN, I decided to move away from the highly unstandard name of ConfigFile and rename the module to Config::File. A small redirecting module is put in place, so current code using this module does not break, but the ConfigFile namespace usage is deprecated (and will thus issue a warning). Please update your code!

### **AUTHOR**

Development was started by Sebastien J. Gross <seb@sjgross.org>, and since 2003 it is maintained by Gunnar Wolf <gwolf@gwolf.org>.

All rights reserved. This program is free software; you can redistribute it and/or modify it under the terms of the GPL.

### **VERSION**

Version 1.4 Copyright (c) 2002 Sebastien J. Gross. All rights reserved. Copyright (c) 2003-2009 Gunnar Wolf. All rights reserved. This program is free software; you can redistribute it and/or modify it under the terms of the GPL v2 (or later, at your choice).