

*This wiki is automatically published from ohmyzsh/wiki. To edit this page, go to ohmyzsh/wiki, make your changes and submit a Pull Request.*

## Settings

**NOTE:** unless specified otherwise, these variables need to be declared *before* Oh My Zsh is sourced in your `.zshrc` file.

- Main settings
  - ZSH
  - ZSH\_THEME
  - plugins
  - ZSH\_CUSTOM
  - ZSH\_CACHE\_DIR
- Update settings
  - `'omz:update'` mode
  - `'omz:update'` frequency
  - Deprecated settings
- Completion settings
  - ZSH\_COMPDUMP
  - ZSH\_DISABLE\_COMPFIX
  - COMPLETION\_WAITING\_DOTS
  - CASE\_SENSITIVE
  - HYPHEN\_INSENSITIVE
- Automatic title
  - DISABLE\_AUTO\_TITLE
  - ZSH\_THEME\_TERM\_TITLE\_IDLE
  - ZSH\_THEME\_TERM\_TAB\_TITLE\_IDLE
- Library settings
  - DISABLE\_MAGIC\_FUNCTIONS
  - DISABLE\_LS\_COLORS
  - ENABLE\_CORRECTION
  - DISABLE\_UNTRACKED\_FILES\_DIRTY
  - HIST\_STAMPS
- Random theme
  - ZSH\_THEME\_RANDOM\_CANDIDATES
  - ZSH\_THEME\_RANDOM\_IGNORED
  - ZSH\_THEME\_RANDOM\_QUIET

For other settings:

- For a plugin, look up the plugin README.
- For a theme, look up the `<theme_name>.zsh-theme` file.

## Main settings

These are the main variables which control Oh My Zsh. Some are required and some are optional; that is specified for each setting.

### ZSH

(*Required*) This variable points to the path of the Oh My Zsh folder. *By default*, Oh My Zsh is installed in `$HOME/.oh-my-zsh`, but if you ran the installer with a different path this will be set accordingly in your `.zshrc` file.

```
export ZSH="$HOME/.oh-my-zsh"
```

It's important that this variable is set, but if it isn't, Oh My Zsh will try to determine its value when loading Oh My Zsh to the directory where the init script (`oh-my-zsh.sh`) is located.

### ZSH\_THEME

(*Optional*) This variable holds the name of the Oh My Zsh you want to use. See `[[Themes]]` for valid theme names, or `[[External themes]]` for themes that aren't included in Oh My Zsh. For example:

```
ZSH_THEME=agnoster
```

If this is not set, Oh My Zsh will not load any themes and you'll get the default zsh prompt.

**NOTE:** if there's a built-in theme and a custom theme of the same name, the custom theme has preference, meaning it will be loaded instead of the built-in one.

### plugins

(*Optional*) (array) This variable is an array containing the plugins that should be loaded when loading Oh My Zsh. See `[[Plugins]]` for valid plugins and the link to their README. Note that, in zsh, array elements are separated by spaces (do not use commas). For example:

```
plugins=(git dircycle autojump)
```

The order of the plugins in the array controls the order in which they'll be loaded. In the example above, the `git` plugin will be loaded first, then the `dircycle` plugin and then the `autojump` plugin.

**NOTE:** as it happens with themes, if there's a custom plugin of the same name as a built-in one, the custom plugin will be loaded instead.

## ZSH\_CUSTOM

(*Optional*) Path to the custom folder. *By default*, this variable points to `$ZSH/custom`. This variable is useful to, for instance, put the custom folder on another location so as to manage it with a version control system. For example:

```
ZSH_CUSTOM=~/.code/ohmyzsh-custom
```

## ZSH\_CACHE\_DIR

(*Optional*) Path to the cache folder. This folder is used to store and cache all sorts of data used by plugins and completion scripts. *By default*, this variable points to `$ZSH/cache`, but you can put it wherever else you see fit. For example:

```
ZSH_CACHE_DIR="${XDG_CACHE_HOME:-$HOME/.cache}/ohmyzsh"
```

## Update settings

### `':omz:update' mode`

This setting controls which automatic update mode to use. These are the available modes:

1. **disabled**: disables all automatic updates.  

```
zstyle ':omz:update' mode disabled
```
2. **auto**: automatically updates Oh My Zsh when a new version is available, without asking for confirmation.  

```
zstyle ':omz:update' mode auto
```
3. **reminder**: only checks if there are updates available and shows a reminder to update Oh My Zsh.  

```
zstyle ':omz:update' mode reminder
```
4. **prompt**: it asks for confirmation before updating Oh My Zsh. This is the default mode, **so you don't need to set it** (just delete the `zstyle` setting if you've changed it before).

### `':omz:update' frequency`

This setting tells Oh My Zsh how often should automatic updates happen (**in days**). This setting only takes effect when automatic updates are enabled. **The default are 13 days.**

```
# Check for updates every 7 days
zstyle ':omz:update' frequency 7
```

## Deprecated settings

These settings are still supported but will be removed in a future version of Oh My Zsh. Migrate to the `zstyle` settings while you still can.

- `DISABLE_AUTO_UPDATE=true`: if set, it has the same effect as setting **disabled mode** in the new `zstyle` format. Instead, use:

```
zstyle ':omz:update' mode disabled
```

- `DISABLE_UPDATE_PROMPT=true`: if set, it has the same effect as setting **auto mode** in the new `zstyle` format. Instead, use:

```
zstyle ':omz:update' mode auto
```

- `UPDATE_ZSH_DAYS=<days>`: if set, it has the same effect as setting **frequency <days>** in the new `zstyle` format. Instead, use:

```
zstyle ':omz:update' frequency <days>
```

## Completion settings

### ZSH\_COMPDUMP

(*Optional*) Path to the completion cache file. *By default*, the name of the cache file is computed using the `$SHORT_HOST` (hostname) and `$ZSH_VERSION` variables, and put in either `$ZDOTDIR` or `$HOME`. You can change it to avoid cluttering your home directory. For example:

```
# If $ZSH_CACHE_DIR is already defined
ZSH_COMPDUMP="$ZSH_CACHE_DIR/.zcompdump"
```

As explained in the [\[\[FAQ\]\]](#), you can reset the completion cache file by `rm`-ing it and restarting the `zsh` session:

```
rm "$ZSH_COMPDUMP"
exec zsh
```

### ZSH\_DISABLE\_COMPFIX

(*Optional*) Zsh automatically detects directories in `$fpath` that might have insecure permissions. These are directories that are checked when loading completion functions or other functions, so if a directory has insecure permissions, it could mean that you end up running code that a malicious actor put there.

**You only need to use this setting when** the directories detected by Zsh have secure permissions but you still get the warning message. By setting this variable to `true`, you won't get the warning anymore. For example:

If you get this or a similar warning in macOS:

```
[oh-my-zsh] Insecure completion-dependent directories detected:
drwxr-xr-x  3 john  admin   96 Jul 25 23:13 /usr/local/share/zsh
drwxr-xr-x  4 john  admin  128 Jul 26 03:38 /usr/local/share/zsh/site-functions
```

you can safely ignore it as long as you control the `john` user, which has write permissions in both those directories. To do that, set the `ZSH_DISABLE_COMPFIX` variable, **before** Oh My Zsh is sourced in your `.zshrc` file:

```
ZSH_DISABLE_COMPFIX=true
```

#### COMPLETION\_WAITING\_DOTS

If you enable this setting, Oh My Zsh will print a red ellipsis to indicate that Zsh is still processing a completion request, and will disappear when the completion finishes. This is useful for example when completing a command that requires a lot of processing before offering completion entries.

```
COMPLETION_WAITING_DOTS=true
```

You can also set it to a string other than `false` so that it is shown instead of the default red ellipsis. For example:

```
COMPLETION_WAITING_DOTS="%F{yellow}waiting...%f"
```

**NOTE: this setting has been found to cause issues with multiline prompt themes (zsh 5.7.1 and newer seem to work).**

#### CASE\_SENSITIVE

Set to `true` to force case-sensitive completion. Otherwise, case-insensitive matching will be applied on filenames. For example, if there are two files beginning with `file`, one lowercase (`file-one`), one uppercase (`FILE-TWO`), the completion system will offer both as entries when trying to complete `file`, unless `CASE_SENSITIVE=true` is applied.

By default, both `file-one` and `FILE-TWO` will match:

```
$ cat file<TAB>
file-one FILE-TWO
```

With `CASE_SENSITIVE=true`, only `file-one` will match:

```
$ cat file<TAB>
$ cat file-one
```

#### HYPHEN\_INSENSITIVE

Case-sensitive completion must be off. Underscores (`_`) and hyphens (`-`) will be interchangeable, if `HYPHEN_INSENSITIVE=true`.

```
$ cat file-<TAB>
file-one file_two
```

## Automatic title

### DISABLE\_AUTO\_TITLE

Oh My Zsh automatically sets the title of your terminal and tabs when running a command or printing the prompt. Use this setting if you want to disable that.

`DISABLE_AUTO_TITLE=true`

### ZSH\_THEME\_TERM\_TITLE\_IDLE

**This variable needs to be set after Oh My Zsh has been sourced.**

Default title for the terminal when the shell is not running a command. This is used just before printing the prompt.

This variable is a string with Prompt Expansion sequences, so you can use any of the prompt sequences defined in that documentation page.

**Default:** shows username, hostname and current directory: `%n@%m:%~` (in Terminal.app the directory is omitted).

### ZSH\_THEME\_TERM\_TAB\_TITLE\_IDLE

**This variable needs to be set after Oh My Zsh has been sourced.**

This is similar to `ZSH_THEME_TERM_TITLE_IDLE` but it targets the terminal tab instead. Note that some terminals might use the terminal title and the terminal tab title interchangeably, so if changing one setting doesn't do it you can try changing the other.

**Default:** current directory truncated to a maximum of 15 characters: `%15<..<%~%<<`.

## Library settings

These settings control the behavior of the library parts of Oh My Zsh. These libraries are located in the `lib/` folder of the project. **Again, these settings need to be set before Oh My Zsh is sourced.**

### DISABLE\_MAGIC\_FUNCTIONS

`bracketed-paste-magic` and `url-quote-magic` are two Zsh utilities that are known buggy in some Zsh versions or user setups. If you're having problems when pasting URLs or pasting anything at all, use this setting to disable them.

`DISABLE_MAGIC_FUNCTIONS=true`

## DISABLE\_LS\_COLORS

Use this setting to disable the Oh My Zsh logic to automatically set `ls` color output based on the system you're running and which `ls` commands are available.

`DISABLE_LS_COLORS=true`

## ENABLE\_CORRECTION

If you use this setting, Oh My Zsh will use `setopt correct_all`, which tells Zsh to try to correct command names and filenames passed as arguments. Only the following commands will be prevented to have filename correction: `cp`, `ebuild`, `gist`, `heroku`, `hpodder`, `man`, `mkdir`, `mv`, `mysql`, `sudo`.

`ENABLE_CORRECTION=true`

## DISABLE\_UNTRACKED\_FILES\_DIRTY

Use this setting if you want to disable marking untracked files under VCS as dirty. This makes repository status checks for large repositories much, much faster.

`DISABLE_UNTRACKED_FILES_DIRTY=true`

NOTE: this setting only takes effect if your theme calls the `git_prompt_info` or `parse_git_dirty` git prompt functions in `lib/git.zsh`.

## HIST\_STAMPS

Oh My Zsh provides a wrapper for the `history` command. You can use this setting to decide whether to show a timestamp for each command in the history output.

Valid values are:

- `"mm/dd/yyyy"`: for `<month>/<day>/<year>` (12/31/2020).
- `"dd.mm.yyyy"`: for `<day>.<month>.<year>` (31.12.2020).
- `"yyyy-mm-dd"`: for `<year>-<month>-<day>` (2020-12-31).
- Custom value: you can specify another format using the `strftime` format (for example, `"%d/%m/%Y"` for 31/12/2020).

Example, if `HIST_STAMPS="dd.mm.yyyy"`:

```
$ history -5
10001  10.10.2020 13:29  gd
10002  10.10.2020 13:29  z oh
10003  10.10.2020 13:29  gd
10004  10.10.2020 13:54  vsc /home/marc/code/ohmyzsh/wiki
10005  10.10.2020 14:36  history -5
```

## Random theme

These settings only work if the random theme is selected (`ZSH_THEME=random`).

### `ZSH_THEME_RANDOM_CANDIDATES`

(Array) If this variable is set, the random theme will choose only one of the themes specified in this array. **Otherwise**, the random theme chooses one from all the themes found in `$ZSH/themes` and `$ZSH_CUSTOM`. For example:

```
ZSH_THEME_RANDOM_CANDIDATES=(robbyrussell af-magic ys)
```

In this example, only 1 of these 3 themes will be selected at random. This is useful when you've used the random theme enough that you know specifically the themes that you like.

**NOTE:** if this variable is set, the `ZSH_THEME_RANDOM_IGNORED` setting has no effect.

### `ZSH_THEME_RANDOM_IGNORED`

(Array) If this variable is set the random theme won't choose any of the themes specified in this array. This is useful if you know specifically the themes that you don't like or don't work correctly in your environment. For example:

```
ZSH_THEME_RANDOM_IGNORED=(agnoster pygmalion rkj)
```

In this example, the random theme will remove these 3 themes from the pool of candidates to choose from.

**NOTE:** if the `ZSH_THEME_RANDOM_CANDIDATES` variable is set (see above), this setting has no effect.

### `ZSH_THEME_RANDOM_QUIET`

If this variable is set to `true`, the random theme will not show a startup message indicating which theme was chosen. For example:

```
ZSH_THEME_RANDOM_QUIET=true
```

If this is set and you want to know which theme was chosen, you can `echo $RANDOM_THEME` to show the theme name.