

zinit-install.zsh - a shell script

Documentation automatically generated with `zshelldoc`

```
__zi::atclone-hook
__zi::atpull-e-hook
__zi::atpull-hook
__zi::compile-plugin-hook
__zi::configure-base-hook
__zi::configure-e-hook
__zi::configure-hook
__zi::cp-hook
__zi::extract-hook
__zi::make-base-hook
__zi::make-e-hook
__zi::make-ee-hook
__zi::make-hook
__zi::mv-hook
__zi::ps-on-update-hook
__zi::reset-hook
zi::at-eval
zi::compile-plugin
zi::compinit
zi::configure-run-autoconf
zi::download-file-stdout
zi::download-snippet
zi::extract
zi::forget-completion
zi::get-cygwin-package
zi::get-latest-gh-r-url-part
zi::get-package
zi::get-url-mtime
zi::install-completions
zi::jq-check
zi::json-get-value
zi::json-to-array
zi::mirror-using-svn
zi::setup-plugin-dir
zi::single-line
zi::update-snippet
zicp
ziextract
zimv
zpextract
AUTOLOAD compinit
```

Has 6 line(s). No functions are called (may set up e.g. a hook, a Zle widget bound to a key, etc.).

Uses feature(s): *source*

Has 26 line(s). Calls functions:

```
__zi::atclone-hook  
|-- zinit-side.zsh/zi::countdown  
'-- zinit.zsh/@zi::substitute
```

Uses feature(s): *eval*, *setopt*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 22 line(s). Calls functions:

```
__zi::atpull-e-hook  
|-- zi::at-eval  
|   '-- zinit.zsh/@zi::substitute  
'-- zinit-side.zsh/zi::countdown
```

Uses feature(s): *setopt*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 22 line(s). Calls functions:

```
__zi::atpull-hook  
|-- zi::at-eval  
|   '-- zinit.zsh/@zi::substitute  
'-- zinit-side.zsh/zi::countdown
```

Uses feature(s): *setopt*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 19 line(s). Calls functions:

```
__zi::compile-plugin-hook
`-- zi::compile-plugin
   |-- zinit-side.zsh/zi::compute-ice
   |-- zinit-side.zsh/zi::first
   `-- zinit.zsh/+zi::message
```

Uses feature(s): *setopt*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

A base common implementation of `configure''`, as all the starting steps are rigid and the same in all hooks, hence the idea. TODO: use in `make''` and other places.

Has 65 line(s). Calls functions:

```
__zi::configure-base-hook
|-- zi::configure-run-autoconf
|  `-- zinit-side.zsh/zi::countdown
|-- zinit-side.zsh/zi::countdown
`-- zinit.zsh/@zi::substitute
```

Called by:

```
__zi::configure-e-hook
__zi::configure-hook
```

Environment variables used: `zinit.zsh` → `ZPFX`

The `!`-version of `configure''` `ice`. Runs in between of `make'!!'` and `make'!'`. Configure naturally runs before `make`.

Has 1 line(s). Calls functions:

```
__zi::configure-e-hook
`-- __zi::configure-base-hook
    |-- zi::configure-run-autoconf
    |   `-- zinit-side.zsh/zi::countdown
    |-- zinit-side.zsh/zi::countdown
    `-- zinit.zsh/@zi::substitute
```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

The non-! version of configure'' ice. Runs in between of make'' and make''. Configure script naturally runs before make.

Has 1 line(s). Calls functions:

```
__zi::configure-hook
`-- __zi::configure-base-hook
    |-- zi::configure-run-autoconf
    |   `-- zinit-side.zsh/zi::countdown
    |-- zinit-side.zsh/zi::countdown
    `-- zinit.zsh/@zi::substitute
```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 30 line(s). Calls functions:

```
__zi::cp-hook
`-- zinit.zsh/@zi::substitute
```

Uses feature(s): *setopt*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 10 line(s). Calls functions:

```

__zi::extract-hook
|-- zi::extract
|   |-- ziextract
|   |   '-- zinit.zsh/+zi::message
|   '-- zinit.zsh/+zi::message
'-- zinit.zsh/@zi::substitute

```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 29 line(s). Calls functions:

```

__zi::make-base-hook
|-- zinit-side.zsh/zi::countdown
'-- zinit.zsh/@zi::substitute

```

Called by:

```

__zi::make-e-hook
__zi::make-ee-hook
__zi::make-hook

```

Has 1 line(s). Calls functions:

```

__zi::make-e-hook
'-- __zi::make-base-hook
    |-- zinit-side.zsh/zi::countdown
    '-- zinit.zsh/@zi::substitute

```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 1 line(s). Calls functions:

```

__zi::make-ee-hook
'-- __zi::make-base-hook
    |-- zinit-side.zsh/zi::countdown
    '-- zinit.zsh/@zi::substitute

```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 1 line(s). Calls functions:

```
__zi::make-hook
`-- __zi::make-base-hook
    |-- zinit-side.zsh/zi::countdown
    `-- zinit.zsh/@zi::substitute
```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 35 line(s). Calls functions:

```
__zi::mv-hook
|-- zinit.zsh/+zi::message
`-- zinit.zsh/@zi::substitute
```

Uses feature(s): *setopt*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 18 line(s). Calls functions:

```
__zi::ps-on-update-hook
`-- zinit.zsh/+zi::message
```

Uses feature(s): *eval*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 79 line(s). Calls functions:

```
__zi::reset-hook
`-- zinit.zsh/+zi::message
```

Uses feature(s): *eval*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 9 line(s). Calls functions:

```
zi::at-eval  
'-- zinit.zsh/@zi::substitute
```

Uses feature(s): *eval*

Called by:

```
__zi::atpull-e-hook  
__zi::atpull-hook
```

Compiles given plugin (its main source file, and also an additional "....zsh" file if it exists).

\$1 - plugin spec (4 formats: user---plugin, user/plugin, user, plugin)
\$2 - plugin (only when \$1 - i.e. user - given)

Has 87 line(s). Calls functions:

```
zi::compile-plugin  
|-- zinit-side.zsh/zi::compute-ice  
|-- zinit-side.zsh/zi::first  
'-- zinit.zsh/+zi::message
```

Uses feature(s): *eval*, *setopt*, *zcompile*

Called by:

```
__zi::compile-plugin-hook  
zinit-autoload.zsh/zi::compile-uncompile-all  
zinit.zsh/zinit
```


User-exposed 'compinit' frontend which first ensures that all completions managed by Zinit are forgotten by Zshell. After that it runs normal 'compinit', which should more easily detect Zinit's completions.

No arguments.

Has 26 line(s). Calls functions:

```
zi::compinit
|-- compinit
|-- zi::forget-completion
`-- zinit.zsh/zi::message
```

Uses feature(s): *autoload, compinit, setopt, unfunction*

Called by:

```
zi::install-completions
zinit-autoload.zsh/zi::uninstall-completions
zinit-autoload.zsh/zi::update-or-status-all
zinit.zsh/zi::prepare-home
zinit.zsh/zinit
```

Called either because "#" flag given to configure'', or because theres no ./configure script. Runs autoconf, autoreconf, and autogen.sh as needed.

Has 64 line(s). Calls functions:

```
zi::configure-run-autoconf
`-- zinit-side.zsh/zi::countdown
```

Called by:

```
__zi::configure-base-hook
```

Downloads file to stdout. Supports following backend commands:
curl, wget, lftp, lynx. Used by snippet loading.

Has 53 line(s). Calls functions:

```
zi::download-file-stdout  
'-- zinit.zsh/+zi::message
```

Uses feature(s): *setopt*, *trap*, *type*

Called by:

```
zi::download-snippet  
zi::get-cygwin-package  
zi::get-package  
zi::setup-plugin-dir
```

Downloads snippet
file `␣` with curl, wget, lftp or lynx,
directory, with Subversion `␣` when svn-ICE is active.

Github supports Subversion protocol and allows to clone subdirectories.
This is used to provide a layer of support for Oh-My-Zsh and Prezto.

Has 377 line(s). Calls functions:

```

zi::download-snippet
|-- zi::download-file-stdout
|   '-- zinit.zsh/+zi::message
|-- zi::get-url-mtime
|-- zi::install-completions
|   |-- zi::compinit
|   |   |-- compinit
|   |   |-- zi::forget-completion
|   |   '-- zinit.zsh/+zi::message
|   |-- zi::forget-completion
|   |-- zinit-side.zsh/zi::any-colorify-as-uspl2
|   |-- zinit-side.zsh/zi::exists-physically-message
|   |-- zinit.zsh/+zi::message
|   '-- zinit.zsh/zi::any-to-user-plugin
|-- zi::mirror-using-svn
|-- zinit-side.zsh/zi::store-ices
|-- zinit.zsh/+zi::message
'-- zinit.zsh/is-at-least

```

Uses feature(s): *is-at-least*, *setopt*, *trap*, *zcompile*

Called by:

```

zi::update-snippet
zinit.zsh/zi::load-snippet

```

Has 30 line(s). Calls functions:

```

zi::extract
|-- ziextract
|   '-- zinit.zsh/+zi::message
'-- zinit.zsh/+zi::message

```

Uses feature(s): *setopt*

Called by:

```

__zi::extract-hook

```

Implements alternation of Zsh state so that already initialized completion stops being visible to Zsh.

\$1 - completion function name, e.g. "_cp"; can also be "cp"

Has 20 line(s). Doesn't call other functions.

Uses feature(s): *setopt*, *unfunction*

Called by:

```
zi::compinit
zi::install-completions
zinit-autoload.zsh/zi::uninstall-completions
zinit.zsh/zinit
```

Has 70 line(s). Calls functions:

```
zi::get-cygwin-package
|-- zi::download-file-stdout
|   '-- zinit.zsh/+zi::message
'-- zinit.zsh/+zi::message
```

Uses feature(s): *setopt*

Called by:

```
zi::setup-plugin-dir
```

Gets version string of latest release of given Github package. Connects to Github releases page.

Has 116 line(s). Calls functions:

```
zi::get-latest-gh-r-url-part
'-- zinit.zsh/+zi::message
```

Uses feature(s): *setopt*

Called by:

```
zi::setup-plugin-dir  
zinit-autoload.zsh/zi::update-or-status
```

Has 195 line(s). Calls functions:

```
zi::get-package  
|-- zi::download-file-stdout  
|   '-- zinit.zsh/+zi::message  
|-- zi::jq-check  
|   '-- zinit.zsh/+zi::message  
|-- zi::json-to-array  
|   '-- zi::jq-check  
|       '-- zinit.zsh/+zi::message  
|-- ziextract  
|   '-- zinit.zsh/+zi::message  
|-- zinit.zsh/+zi::message  
'-- zinit.zsh/@zi::substitute
```

Uses feature(s): *eval, setopt, trap*

Called by:

```
zinit.zsh/zi::load
```

Environment variables used: zinit.zsh → ZPFX

For the given URL returns the date in the Last-Modified header as a time stamp

Has 35 line(s). Doesn't call other functions.

Uses feature(s): *read, setopt, trap, type*

Called by:

```
zi::download-snippet
```

Installs all completions of given plugin. After that they are visible to 'compinit'. Visible completions can be selectively disabled and enabled. User can access completion data with 'clist' or 'completions' subcommand.

\$1 - plugin spec (4 formats: user---plugin, user/plugin, user, plugin)
\$2 - plugin if \$1 (i.e., user) given
\$3 - if 1, then reinstall, otherwise only install completions that are not present

Has 61 line(s). Calls functions:

```
zi::install-completions
|-- zi::compinit
|   |-- compinit
|   |-- zi::forget-completion
|   `-- zinit.zsh/+zi::message
|-- zi::forget-completion
|-- zinit-side.zsh/zi::any-colorify-as-uspl2
|-- zinit-side.zsh/zi::exists-physically-message
|-- zinit.zsh/+zi::message
`-- zinit.zsh/zi::any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
zi::download-snippet
zi::setup-plugin-dir
zinit.zsh/zinit
```

Check if jq is available and outputs an error message with instructions if that's not the case

Has 8 line(s). Calls functions:

```
zi::jq-check
`-- zinit.zsh/+zi::message
```

Called by:

```
zi::get-package  
zi::json-get-value  
zi::json-to-array
```

Wrapper around jq that return the value of a property

```
$1: JSON structure  
$2: jq path
```

Has 4 line(s). Calls functions:

```
zi::json-get-value  
'-- zi::jq-check  
    '-- zinit.zsh/+zi::message
```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Wrapper around jq that sets key/values of an associative array, replicating the structure of a given JSON object

```
$1: JSON structure  
$2: jq path  
$3: name of the associative array to store the key/value pairs in
```

Has 13 line(s). Calls functions:

```
zi::json-to-array  
'-- zi::jq-check  
    '-- zinit.zsh/+zi::message
```

Uses feature(s): *eval*, *setopt*

Called by:

zi::get-package

Used to clone subdirectories from Github. If in update mode (see \$2), then invokes 'svn update', in normal mode invokes 'svn checkout --non-interactive -q <URL>'. In test mode only compares remote and local revision and outputs true if update is needed.

\$1 - URL
\$2 - mode, "" - normal, "-u" - update, "-t" - test
\$3 - subdirectory (not path) with working copy, needed for -t and -u

Has 29 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

zi::download-snippet

Clones given plugin into PLUGIN_DIR. Supports multiple sites (respecting 'from' and 'proto' ice modifiers). Invokes compilation of plugin's main file.

\$1 - user
\$2 - plugin

Has 215 line(s). Calls functions:


```

zi::setup-plugin-dir
|-- zi::download-file-stdout
|   '-- zinit.zsh/zi::message
|-- zi::get-cygwin-package
|   |-- zi::download-file-stdout
|   |   '-- zinit.zsh/zi::message
|   '-- zinit.zsh/zi::message
|-- zi::get-latest-gh-r-url-part
|   '-- zinit.zsh/zi::message
|-- zi::install-completions
|   |-- zi::compinit
|   |   |-- compinit
|   |   |-- zi::forget-completion
|   |   '-- zinit.zsh/zi::message
|   |-- zi::forget-completion
|   |-- zinit-side.zsh/zi::any-colorify-as-uspl2
|   |-- zinit-side.zsh/zi::exists-physically-message
|   |-- zinit.zsh/zi::message
|   '-- zinit.zsh/zi::any-to-user-plugin
|-- ziextract
|   '-- zinit.zsh/zi::message
|-- zinit-side.zsh/zi::any-colorify-as-uspl2
|-- zinit-side.zsh/zi::store-ices
|-- zinit.zsh/zi::message
'-- zinit.zsh/zi::get-object-path

```

Uses feature(s): *setopt*, *trap*

Called by:

```

zinit-autoload.zsh/zi::update-or-status
zinit.zsh/zi::load

```

Display cURL progress bar on a single line

Has 20 line(s). Doesn't call other functions.

Uses feature(s): *read*, *setopt*

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 76 line(s). Calls functions:

```

zi::update-snippet
|-- zi::download-snippet
|   |-- zi::download-file-stdout
|   |   '-- zinit.zsh/+zi::message
|   |-- zi::get-url-mtime
|   |-- zi::install-completions
|   |   |-- zi::compinit
|   |   |   |-- compinit
|   |   |   |-- zi::forget-completion
|   |   |   '-- zinit.zsh/+zi::message
|   |   |-- zi::forget-completion
|   |   |-- zinit-side.zsh/zi::any-colorify-as-uspl2
|   |   |-- zinit-side.zsh/zi::exists-physically-message
|   |   |-- zinit.zsh/+zi::message
|   |   '-- zinit.zsh/zi::any-to-user-plugin
|   |-- zi::mirror-using-svn
|   |-- zinit-side.zsh/zi::store-ices
|   |-- zinit.zsh/+zi::message
|   '-- zinit.zsh/is-at-least
|-- zinit.zsh/+zi::message
|-- zinit.zsh/zi::get-object-path
'-- zinit.zsh/zi::pack-ice

```

Uses feature(s): *eval*, *setopt*

Called by:

```
zinit-autoload.zsh/zi::update-or-status-snippet
```

Has 30 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
zimv
```

Environment variables used: *zinit.zsh* → *ZPFX*

If the file is an archive, it is extracted by this function.
Next stage is scanning of files with the common utility 'file',
to detect executables. They are given +x mode. There are also
messages to the user on performed actions.

\$1 - url
\$2 - file

Has 283 line(s). Calls functions:

```
ziextract  
'-- zinit.zsh/+zi::message
```

Uses feature(s): *setopt, unfunction, zparseopts*

Called by:

```
zi::extract  
zi::get-package  
zi::setup-plugin-dir  
zpextract
```

Has 3 line(s). Calls functions:

```
zimv  
'-- zicp
```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Has 1 line(s). Calls functions:

```
zpextract  
'-- ziextract  
    '-- zinit.zsh/+zi::message
```

Not called by script or any function (may be e.g. a hook, a Zle widget, etc.).

Initialisation for new style completion. This mainly contains some helper functions and setup. Everything else is split into different files that will automatically be made autoloaded (see the end of this file). The names of the files that will be considered for autoloading are those that begin with an underscores (like ``_condition``).

The first line of each of these files is read and must indicate what should be done with its contents:

```
`#compdef <names ...>'
```

Has 573 line(s). Doesn't call other functions.

Uses feature(s): *autoload*, *bindkey*, *compdef*, *compdump*, *eval*, *read*, *setopt*, *unfunction*, *zle*, *zstyle*

Called by:

```
zi::compinit
```