zi.zsh(1)

NAME

zi.zsh - a shell script

SYNOPSIS

Documentation automatically generated with `zsdoc'

FUNCTIONS

```
+zi-deploy-message
+zi-message
+zi-prehelp-usage-message
-zi_scheduler_add_sh
.zi-add-fpath
.zi-add-report
.zi-any-to-pid
.zi-any-to-user-plugin
.zi-compdef-clear
.zi-compdef-replay
.zi-diff
.zi-diff-env
.zi-diff-functions
.zi-diff-options
.zi-diff-parameter
.zi-find-other-matches
.zi-formatter-auto
.zi-formatter-bar
.zi-formatter-bar-util
.zi-formatter-pid
.zi-formatter-th-bar
.zi-formatter-url
.zi-get-mtime-into
.zi-get-object-path
.zi-ice
.zi-load
.zi-load-ices
.zi-load-object
.zi-load-plugin
.zi-load-snippet
.zi-main-message-formatter
.zi-pack-ice
.zi-parse-opts
.zi-prepare-home
```

```
.zi-register-plugin
.zi-run
.zi-run-task
.zi-set-m-func
.zi-setup-params
.zi-submit-turbo
.zi-tmp-subst-off
.zi-tmp-subst-on
.zi-util-shands-path
:zi-reload-and-run
:zi-tmp-subst-alias
:zi-tmp-subst-autoload
:zi-tmp-subst-bindkey
:zi-tmp-subst-compdef
:zi-tmp-subst-zle
:zi-tmp-subst-zstyle
@autoload
@zi-register-annex
@zi-register-hook
@zi-substitute
@zsh-plugin-run-on-unload
@zsh-plugin-run-on-update
pmodload
zi
zi-turbo
zicdclear
zicdreplay
zicompdef
zicompinit
zicompinit_fast
zpcdclear
zpcdreplay
zpcompdef
zpcompinit
▒▒▒
AUTOLOAD add-zsh-hook
AUTOLOAD compinit
AUTOLOAD is-at-least
PRECMD-HOOK @zi-scheduler
```

DETAILS

Script Body

Has 279 line(s). Calls functions:

```
Script-Body
|-- @zi-register-hook
```

```
|-- add-zsh-hook
|-- is-at-least
`-- side.zsh/.zi-check-module
```

Uses feature(s): *add-zsh-hook, alias, autoload, is-at-least, setopt, source, zmodload, zstyle*

+zi-deploy-message

```
FUNCTION: +zi-deploy-message. [[[
Deploys a sub-prompt message to be displayed OR a `zle
.reset-prompt' call to be invoked
```

Has 13 line(s). Doesn't call other functions.

Uses feature(s): *read, zle*

Called by:

```
.zi-load-snippet
.zi-load
autoload.zsh/.zi-recall
```

+zi-message

```
FUNCTION: +zi-message. [[[
```

Has 13 line(s). Doesn't call other functions.

Called by:

```
+zi-prehelp-usage-message
.zi-compdef-clear
.zi-compdef-replay
.zi-load-snippet
.zi-register-plugin
.zi-run
.zi-set-m-func
:zi-tmp-subst-autoload
zi
additional.zsh/.zi-debug-start
additional.zsh/.zi-debug-unload
additional.zsh/:zi-tmp-subst-source
autoload.zsh/.zi-analytics-menu
autoload.zsh/.zi-build-module
```

```
autoload.zsh/.zi-cd
autoload.zsh/.zi-clear-completions
autoload.zsh/.zi-compiled
autoload.zsh/.zi-glance
autoload.zsh/.zi-help
autoload.zsh/.zi-module
autoload.zsh/.zi-registered-ice-mods
autoload.zsh/.zi-registered-subcommands
autoload.zsh/.zi-self-update
autoload.zsh/.zi-show-report
autoload.zsh/.zi-show-zstatus
autoload.zsh/.zi-uncompile-plugin
autoload.zsh/.zi-uninstall-completions
autoload.zsh/.zi-unload
autoload.zsh/.zi-update-all-parallel
autoload.zsh/.zi-update-or-status-all
autoload.zsh/.zi-update-or-status
autoload.zsh/.zi-wait-for-update-jobs
install.zsh/.zi-compile-plugin
install.zsh/.zi-compinit
install.zsh/.zi-download-file-stdout
install.zsh/.zi-download-snippet
install.zsh/.zi-extract
install.zsh/.zi-get-cygwin-package
install.zsh/.zi-get-latest-gh-r-url-part
install.zsh/.zi-get-package
install.zsh/.zi-install-completions
install.zsh/.zi-mirror-using-svn
install.zsh/.zi-setup-plugin-dir
install.zsh/.zi-update-snippet
install.zsh/ziextract
install.zsh/zi-mv-hook
install.zsh/zi-ps-on-update-hook
install.zsh/zi-reset-hook
side.zsh/.zi-check-module
side.zsh/.zi-countdown
side.zsh/.zi-exists-physically-message
```

+zi-prehelp-usage-message

```
FUNCTION: +zi-prehelp-usage-message. [[[
Prints the usage message.
```

Has 34 line(s). Calls functions:

```
+zi-prehelp-usage-message
`-- +zi-message
```

Called by:

```
zi
autoload.zsh/.zi-delete
```

-zi_scheduler_add_sh

```
FUNCTION: -zi_scheduler_add_sh. [[[
Copies task into ZI_RUN array, called when a task timeouts.
A small function ran from pattern in /-substitution as a math
function.
```

Has 7 line(s). Doesn't call other functions.

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

zi-add-fpath

~~~~~

```
FUNCTION: .zi-add-fpath. [[[
```

Has 10 line(s). Calls functions:

```
.zi-add-fpath
```

Called by:

```
zi
```

zi-add-report

~~~~~~

```
FUNCTION: .zi-add-report. [[[
Adds a report line for given plugin.
```

```
$1 - uspl2, i.e. user/plugin
$2, ... - the text
```

Has 3 line(s). Doesn't call other functions.

Called by:

```
.zi-load-plugin
.zi-load-snippet
:zi-tmp-subst-alias
:zi-tmp-subst-autoload
:zi-tmp-subst-bindkey
:zi-tmp-subst-compdef
:zi-tmp-subst-zle
:zi-tmp-subst-zstyle
```

zi-any-to-pid

~~~~~~

```
FUNCTION: .zi-any-to-pid. [[[
```

Has 21 line(s). Calls functions:

```
.zi-any-to-pid
```

Uses feature(s): *setopt*

Called by:

```
side.zsh/.zi-any-colorify-as-uspl2
side.zsh/.zi-exists-physically-message
side.zsh/.zi-first
```

zi-any-to-user-plugin

~~~~~~~

```
FUNCTION: .zi-any-to-user-plugin. [[[
Allows elastic plugin-spec across the code.
```

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

```
Returns user and plugin in $reply.
```

Has 25 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
.zi-add-fpath
.zi-get-object-path
.zi-load
.zi-run
:zi-tmp-subst-autoload
autoload.zsh/.zi-any-to-uspl2
autoload.zsh/.zi-changes
autoload.zsh/.zi-compile-uncompile-all
autoload.zsh/.zi-compiled
autoload.zsh/.zi-create
autoload.zsh/.zi-delete
autoload.zsh/.zi-find-completions-of-plugin
autoload.zsh/.zi-glance
autoload.zsh/.zi-show-report
autoload.zsh/.zi-stress
autoload.zsh/.zi-uncompile-plugin
autoload.zsh/.zi-unload
autoload.zsh/.zi-unregister-plugin
autoload.zsh/.zi-update-all-parallel
autoload.zsh/.zi-update-or-status-all
autoload.zsh/.zi-update-or-status
install.zsh/.zi-install-completions
side.zsh/.zi-any-colorify-as-uspl2
side.zsh/.zi-compute-ice
side.zsh/.zi-exists-physically-message
side.zsh/.zi-exists-physically
side.zsh/.zi-first
```

zi-compdef-clear

~~~~~~~

```
FUNCTION: .zi-compdef-clear. [[[
Implements user-exposed functionality to clear gathered compdefs.
```

Has 3 line(s). Calls functions:

```
.zi-compdef-clear
`-- +zi-message
```

Called by:

```
zi
```

```
zicdclear
zpcdclear
```

zi-compdef-replay

~~~~~~

```
FUNCTION: .zi-compdef-replay. [[[
Runs gathered compdef calls. This allows to run `compinit' after loading plugins.
```

Has 16 line(s). Calls functions:

```
.zi-compdef-replay
`-- +zi-message
```

Uses feature(s): *compdef*

Called by:

```
zi
zicdreplay
zpcdreplay
```

zi-diff

~~~~

```
FUNCTION: .zi-diff. [[[
Performs diff actions of all types
```

Has 4 line(s). Calls functions:

```
.zi-diff
```

Called by:

```
.zi-load-plugin
additional.zsh/.zi-debug-start
additional.zsh/.zi-debug-stop
```

zi-diff-env

~~~~

```
FUNCTION: .zi-diff-env. [[[
```

```
Implements detection of change in PATH and FPATH.
```

```
$1 - user/plugin (i.e. uspl2 format)
$2 - command, can be "begin" or "end"
```

Has 15 line(s). Doesn't call other functions.

Called by:

```
.zi-diff
.zi-load-plugin
```

zi-diff-functions

~~~~~~

```
FUNCTION: .zi-diff-functions. [[[
Implements detection of newly created functions. Performs data gathering,
computation is done in *-compute().
```

```
$1 - user/plugin (i.e. uspl2 format)
$2 - command, can be "begin" or "end"
```

Has 3 line(s). Doesn't call other functions.

Called by:

```
.zi-diff
```

zi-diff-options

~~~~~~

```
FUNCTION: .zi-diff-options. [[[
Implements detection of change in option state. Performs data gathering,
computation is done in *-compute().
```

```
$1 - user/plugin (i.e. uspl2 format)
$2 - command, can be "begin" or "end"
```

Has 2 line(s). Doesn't call other functions.

Called by:

```
.zi-diff
```

zi-diff-parameter

~~~~~

```
FUNCTION: .zi-diff-parameter. [[[
Implements detection of change in any parameter's existence and type.
Performs data gathering, computation is done in *-compute().
```

```
$1 - user/plugin (i.e. uspl2 format)
$2 - command, can be "begin" or "end"
```

Has 6 line(s). Doesn't call other functions.

Called by:

```
.zi-diff
```

zi-find-other-matches

~~~~~~~~

```
FUNCTION: .zi-find-other-matches. [[[
Plugin's main source file is in general `name.plugin.zsh'. However,
there can be different conventions, if that file is not found, then
this functions examines other conventions in the most sane order.
```

Has 17 line(s). Doesn't call other functions.

Called by:

```
.zi-load-plugin
.zi-load-snippet
side.zsh/.zi-first
```

zi-formatter-auto

~~~~~

```
FUNCTION: .zi-formatter-auto[[[
The automatic message formatting tool automatically detects,
formats, and colorizes the following pieces of text:
[URLs], [plugin IDs (user/repo) if exists on the disk], [numbers], [time],
[single-word id-as plugins],
```

```
[single word commands], [single word functions], [zi commands], [ice modifiers
(e.g: mv'a -> b')],
[single-char bits and quoted strings (`...', ,'...', "...")].
```

Has 46 line(s). Calls functions:

```
.zi-formatter-auto
```

Uses feature(s): *setopt*

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

zi-formatter-bar

~~~~~~~

```
FUNCTION: .zi-formatter-bar. [[[
```

Has 1 line(s). Calls functions:

```
.zi-formatter-bar
```

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

zi-formatter-bar-util

~~~~~~~~

```
FUNCTION: .zi-formatter-bar-util. [[[
```

Has 6 line(s). Doesn't call other functions.

Called by:

```
.zi-formatter-bar
.zi-formatter-th-bar
```

zi-formatter-pid

~~~~~~~

```
FUNCTION: .zi-formatter-pid. [[[
```

Has 10 line(s). Calls functions:

```
.zi-formatter-pid
`-- side.zsh/.zi-any-colorify-as-uspl2
```

Uses feature(s): *source*

Called by:

```
.zi-formatter-auto
```

zi-formatter-th-bar

~~~~~~~~

```
FUNCTION: .zi-formatter-th-bar. [[[
```

Has 1 line(s). Calls functions:

```
.zi-formatter-th-bar
```

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

zi-formatter-url

~~~~~~~~

```
FUNCTION: .zi-formatter-url. [[[
```

Has 14 line(s). Doesn't call other functions.

Called by:

```
.zi-formatter-auto
```

zi-get-mtime-into

~~~~~~

```
FUNCTION: .zi-get-mtime-into. [[[
```

Has 7 line(s). Doesn't call other functions.

Called by:

```
Script-Body
```

```
autoload.zsh/.zi-self-update
autoload.zsh/.zi-update-or-status-all
```

zi-get-object-path

~~~~~~~

```
FUNCTION: .zi-get-object-path. [[[
```

Has 23 line(s). Calls functions:

```
.zi-get-object-path
```

Called by:

```
.zi-load-ices
.zi-load-snippet
.zi-run
zi
autoload.zsh/.zi-get-path
install.zsh/.zi-setup-plugin-dir
install.zsh/.zi-update-snippet
side.zsh/.zi-first
side.zsh/.zi-two-paths
```

zi-ice

~~~

```
FUNCTION: .zi-ice. [[[
Parses ICE specification, puts the result into ICE global hash. The ice-spec is
valid for
next command only (i.e. it "melts"), but it can then stick to plugin and activate
e.g. at update.
```

Has 12 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
zi
```

zi-load

~~~~

```
FUNCTION: .zi-load. [[[
Implements the exposed-to-user action of loading a plugin.
```

```
$1 - plugin spec (4 formats: user---plugin, user/plugin, user, plugin)
$2 - plugin name, if the third format is used
```

Has 78 line(s). Calls functions:

```
.zi-load
|-- +zi-deploy-message
|-- install.zsh/.zi-get-package
`-- install.zsh/.zi-setup-plugin-dir
```

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

Called by:

```
.zi-load-object
.zi-run-task
additional.zsh/.zi-service
```

zi-load-ices

~~~~~

```
FUNCTION: .zi-load-ices. [[[
```

Has 22 line(s). Calls functions:

```
.zi-load-ices
```

Called by:

```
zi
```

zi-load-object

~~~~~

```
FUNCTION: .zi-load-object. [[[
```

Has 10 line(s). Calls functions:

```
.zi-load-object
```

Called by:

```
zi
```

zi-load-plugin

~~~~~

```
FUNCTION: .zi-load-plugin. [[[
Lower-level function for loading a plugin.
```

```
$1 - user
$2 - plugin
$3 - mode (light or load)
```

Has 107 line(s). Calls functions:

```
.zi-load-plugin
|-- :zi-tmp-subst-autoload
|   |-- +zi-message
|   `-- is-at-least
`-- additional.zsh/.zi-wrap-functions
```

Uses feature(s): *eval, setopt, source, unfunction, zle*

Called by:

```
.zi-load
```

zi-load-snippet

~~~~~~

```
FUNCTION: .zi-load-snippet. [[[
Implements the exposed-to-user action of loading a snippet.
```

```
$1 - url (can be local, absolute path).
```

Has 173 line(s). Calls functions:

```
.zi-load-snippet
|-- +zi-deploy-message
|-- +zi-message
|-- additional.zsh/.zi-wrap-functions
`-- install.zsh/.zi-download-snippet
```

Uses feature(s): *autoload, eval, setopt, source, unfunction, zparseopts, zstyle*

Called by:

```
.zi-load-object
.zi-load
.zi-run-task
pmodload
additional.zsh/.zi-service
```

zi-main-message-formatter

~~~~~~~~~~~~

```
FUNCTION: +zi-message-formatter [[[
```

Has 18 line(s). Doesn't call other functions.

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

zi-pack-ice

~~~~

```
FUNCTION: .zi-pack-ice. [[[
Remembers all ice-mods, assigns them to concrete plugin. Ice spec is in general
forgotten for
second-next command (that's why it's called "ice" - it melts), however they glue
to the object (plugin
or snippet) mentioned in the next command  for later use with e.g. `zi update
...'.
```

Has 3 line(s). Doesn't call other functions.

Called by:

```
.zi-load-snippet
.zi-load
@zsh-plugin-run-on-unload
@zsh-plugin-run-on-update
install.zsh/.zi-update-snippet
```

```
side.zsh/.zi-compute-ice
```

zi-parse-opts

~~~~~~~

```
]]]
FUNCTION: +zi-parse-opts. [[[
```

Has 2 line(s). Doesn't call other functions.

Called by:

```
zi
autoload.zsh/.zi-delete
```

zi-prepare-home

~~~~~~

```
FUNCTION: .zi-prepare-home. [[[
Establish all required directories.
```

Has 64 line(s). Calls functions:

```
.zi-prepare-home
|-- autoload.zsh/.zi-clear-completions
`-- install.zsh/.zi-compinit
```

Uses feature(s): *source*

Called by:

```
Script-Body
```

zi-register-plugin

~~~~~~~

```
FUNCTION: .zi-register-plugin. [[[
Adds the plugin to ZI_REGISTERED_PLUGINS array and to the
zsh_loaded_plugins array (managed according to the plugin standard:
https://z.digitalclouds.dev/community/zsh_plugin_standard).
```

Has 20 line(s). Calls functions:

```
.zi-register-plugin
`-- +zi-message
```

Called by:

```
.zi-load
```

zi-run

~~~

```
FUNCTION: .zi-run. [[[
Run code inside plugin's folder
It uses the `correct' parameter from upper's scope zi().
```

Has 24 line(s). Calls functions:

```
.zi-run
`-- +zi-message
```

Uses feature(s): *eval, setopt*

Called by:

```
zi
```

zi-run-task

~~~~

```
FUNCTION: .zi-run-task. [[[
A backend, worker function of .zi-scheduler. It obtains the tasks
index and a few of its properties (like the type: plugin, snippet,
service plugin, service snippet) and executes it first checking for
additional conditions (like non-numeric wait'' ice).
```

```
$1 - the pass number, either 1st or 2nd pass
$2 - the time assigned to the task
$3 - type: plugin, snippet, service plugin, service snippet
$4 - task's index in the ZI[WAIT_ICE_...] fields
$5 - mode: load or light
$6 - the plugin-spec or snippet URL or alias name (from id-as'')
```

Has 45 line(s). Calls functions:

```
.zi-run-task
`-- autoload.zsh/.zi-unload
```

Uses feature(s): *eval, source, zle, zpty*

Called by:

```
@zi-scheduler
```

zi-set-m-func

~~~~~~

```
FUNCTION:.zi-set-m-func() [[[
Sets and withdraws the temporary, atclone/atpull time function `m`.
```

Has 17 line(s). Calls functions:

```
.zi-set-m-func
`-- +zi-message
```

Uses feature(s): *setopt*

Called by:

```
.zi-load-snippet
.zi-load
autoload.zsh/.zi-update-or-status
```

zi-setup-params

~~~~~~

```
FUNCTION: .zi-setup-params. [[[
```

Has 3 line(s). Doesn't call other functions.

Called by:

```
.zi-load-snippet
.zi-load
```

zi-submit-turbo

~~~~~~

```
FUNCTION: .zi-submit-turbo. [[[
If `zi load`, `zi light` or `zi snippet`  will be
preceded with `wait', `load', `unload' or `on-update-of`/`subscribe'
ice-mods then the plugin or snipped is to be loaded in turbo-mode,
and this function adds it to internal data structures, so that
@zi-scheduler can run (load, unload) this as a task.
```

Has 16 line(s). Doesn't call other functions.

Called by:

```
zi
```

zi-tmp-subst-off

~~~~~~~~

```
FUNCTION: .zi-tmp-subst-off. [[[
Turn off temporary substituting of functions completely for a given mode ("load",
"light",
"light-b" (i.e. the `trackbinds' mode) or "compdef").
```

Has 17 line(s). Doesn't call other functions.

Uses feature(s): *setopt, unfunction*

Called by:

```
.zi-load-plugin
additional.zsh/.zi-debug-stop
```

zi-tmp-subst-on

~~~~~~

```
FUNCTION: .zi-tmp-subst-on. [[[
Turn on temporary substituting of functions of builtins and functions according
to passed
mode ("load", "light", "light-b" or "compdef"). The temporary substituting of
functions is
to gather report data, and to hijack `autoload', `bindkey' and `compdef' calls.
```

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

Uses feature(s): *source*

Called by:

```
.zi-load-plugin
additional.zsh/.zi-debug-start
```

zi-util-shands-path

~~~~~~~~

```
FUNCTION: .zi-util-shands-path. [[[
Replaces parts of path with %HOME, etc.
```

Has 8 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
.zi-any-to-pid
```

:zi-reload-and-run

```
FUNCTION: :zi-reload-and-run. [[[
Marks given function ($3) for autoloading, and executes it triggering the load.
$1 is the fpath dedicated  to the function, $2 are autoload options. This
function replaces "autoload -X",
because using that on older Zsh versions causes problems with traps.
```

```
So basically one creates function stub that calls :zi-reload-and-run() instead of
"autoload -X".
```

```
$1 - FPATH dedicated to function
$2 - autoload options
$3 - function name (one that needs autoloading)
```

```
Author: Bart Schaefer
```

Has 9 line(s). Doesn't call other functions.

Uses feature(s): *autoload, unfunction*

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

:zi-tmp-subst-alias

```
FUNCTION: :zi-tmp-subst-alias. [[[
Function defined to hijack plugin's calls to the `alias' builtin.
```

```
The hijacking is to gather report data (which is used in unload).
```

Has 30 line(s). Calls functions:

```
:zi-tmp-subst-alias
```

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

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

:zi-tmp-subst-autoload

```
FUNCTION: :zi-tmp-subst-autoload. [[[
Function defined to hijack plugin's calls to the `autoload' builtin.
```

```
The hijacking is not only to gather report data, but also to.
run custom `autoload' function, that doesn't need FPATH.
```

Has 105 line(s). Calls functions:

```
:zi-tmp-subst-autoload
|-- +zi-message
`-- is-at-least
```

Uses feature(s): *autoload, eval, is-at-least, setopt, zparseopts*

Called by:

```
.zi-load-plugin
@autoload
```

:zi-tmp-subst-bindkey

```
FUNCTION: :zi-tmp-subst-bindkey. [[[
Function defined to hijack plugin's calls to the `bindkey' builtin.
```

```
The hijacking is to gather report data (which is used in unload).
```

Has 106 line(s). Calls functions:

```
:zi-tmp-subst-bindkey
`-- is-at-least
```

Uses feature(s): *bindkey, is-at-least, setopt, zparseopts*

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

:zi-tmp-subst-compdef

```
FUNCTION: :zi-tmp-subst-compdef. [[[
Function defined to hijack plugin's calls to the `compdef' function.
The hijacking is not only for reporting, but also to save compdef
calls so that `compinit' can be called after loading plugins.
```

Has 5 line(s). Calls functions:

```
:zi-tmp-subst-compdef
```

Uses feature(s): *setopt*

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

:zi-tmp-subst-zle

```
FUNCTION: :zi-tmp-subst-zle. [[[.
Function defined to hijack plugin's calls to the `zle' builtin.
```

```
The hijacking is to gather report data (which is used in unload).
```

Has 33 line(s). Calls functions:

```
:zi-tmp-subst-zle
```

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

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

:zi-tmp-subst-zstyle

```
FUNCTION: :zi-tmp-subst-zstyle. [[[
Function defined to hijack plugin's calls to the `zstyle' builtin.
```

```
The hijacking is to gather report data (which is used in unload).
```

Has 19 line(s). Calls functions:

```
:zi-tmp-subst-zstyle
```

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

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

@autoload

```
]]]
FUNCTION: @autoload. [[[
```

Has 1 line(s). Calls functions:

```
@autoload
`-- :zi-tmp-subst-autoload
    |-- +zi-message
    `-- is-at-least
```

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

@zi-register-annex

```
FUNCTION: @zi-register-annex. [[[
Registers the z-annex inside Zi  i.e. an Zi extension
```

Has 8 line(s). Doesn't call other functions.

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

@zi-register-hook

```
FUNCTION: @zi-register-hook. [[[
Registers the z-annex inside Zi.
```

Has 6 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
Script-Body
```

@zi-scheduler

```
FUNCTION: @zi-scheduler. [[[
Searches for timeout tasks, executes them. There's an array of tasks
waiting for execution, this scheduler manages them, detects which ones
should be run at current moment, decides to remove (or not) them from
the array after execution.
```

```
$1 - if "following", then it is non-first (second and more)
invocation of the scheduler; this results in chain of `sched'
invocations that results in repetitive @zi-scheduler activity.
```

```
if "burst", then all tasks are marked timeout and executed one
by one; this is handy if e.g. a docker image starts up and
needs to install all turbo-mode plugins without any hesitation
(delay), i.e. "burst" allows to run package installations from
script, not from prompt.
```

Has 72 line(s). **Is a precmd hook**. Calls functions:

```
@zi-scheduler
`-- add-zsh-hook
```

Uses feature(s): *add-zsh-hook, sched, zle*

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

@zi-substitute

```
FUNCTION: @zi-substitute. [[[
```

Has 36 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
autoload.zsh/.zi-at-eval
install.zsh/.zi-at-eval
install.zsh/.zi-get-package
install.zsh/zi-atclone-hook
install.zsh/zi-cp-hook
install.zsh/zi-extract-hook
install.zsh/zi-make-e-hook
install.zsh/zi-make-ee-hook
install.zsh/zi-make-hook
install.zsh/zi-mv-hook
```

@zsh-plugin-run-on-unload

```
FUNCTION: @zsh-plugin-run-on-update. [[[
The Plugin Standard required mechanism, see:
https://z.digitalclouds.dev/community/zsh_plugin_standard
```

Has 2 line(s). Calls functions:

```
@zsh-plugin-run-on-unload
```

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

@zsh-plugin-run-on-update

```
FUNCTION: @zsh-plugin-run-on-update. [[[
The Plugin Standard required mechanism
```

Has 2 line(s). Calls functions:

```
@zsh-plugin-run-on-update
```

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

pmodload

```
FUNCTION: pmodload. [[[
Compatibility with Prezto. Calls can be recursive.
```

Has 15 line(s). Calls functions:

```
pmodload
```

Uses feature(s): *zstyle*

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

zi

```
FUNCTION: zi. [[[
Main function directly exposed to user, obtains subcommand and its arguments, has
completion.
```

Has 550 line(s). Calls functions:

```
zi
|-- +zi-message
|-- +zi-prehelp-usage-message
|    `-- +zi-message
|-- additional.zsh/.zi-clear-debug-report
|-- additional.zsh/.zi-debug-start
|-- additional.zsh/.zi-debug-stop
|-- additional.zsh/.zi-debug-unload
|-- autoload.zsh/.zi-analytics-menu
|-- autoload.zsh/.zi-cdisable
|-- autoload.zsh/.zi-cenable
|-- autoload.zsh/.zi-clear-completions
|-- autoload.zsh/.zi-compile-uncompile-all
|-- autoload.zsh/.zi-compiled
|-- autoload.zsh/.zi-help
|-- autoload.zsh/.zi-list-bindkeys
|-- autoload.zsh/.zi-list-compdef-replay
|-- autoload.zsh/.zi-ls
```

```
|-- autoload.zsh/.zi-module
|-- autoload.zsh/.zi-recently
|-- autoload.zsh/.zi-registered-ice-mods
|-- autoload.zsh/.zi-registered-subcommands
|-- autoload.zsh/.zi-search-completions
|-- autoload.zsh/.zi-self-update
|-- autoload.zsh/.zi-show-all-reports
|-- autoload.zsh/.zi-show-completions
|-- autoload.zsh/.zi-show-debug-report
|-- autoload.zsh/.zi-show-registered-plugins
|-- autoload.zsh/.zi-show-report
|-- autoload.zsh/.zi-show-times
|-- autoload.zsh/.zi-show-zstatus
|-- autoload.zsh/.zi-uncompile-plugin
|-- autoload.zsh/.zi-uninstall-completions
|-- autoload.zsh/.zi-unload
|-- autoload.zsh/.zi-update-or-status
|-- autoload.zsh/.zi-update-or-status-all
|-- compinit
|-- install.zsh/.zi-compile-plugin
|-- install.zsh/.zi-compinit
|-- install.zsh/.zi-forget-completion
`-- install.zsh/.zi-install-completions
```

Uses feature(s): *autoload, compinit, eval, setopt, source*

Called by:

```
zi-turbo
```

zi-turbo

```
FUNCTION: zi-turbo. [[[
With zi-turbo first argument is a wait time and suffix, i.e. "0a".
Anything that doesn't match will be passed as if it were an ice mod.
Default ices depth'3' and lucid, allowed values [0-9][a-c].
```

Has 1 line(s). Calls functions:

```
zi-turbo
`-- zi
    |-- +zi-message
    |-- +zi-prehelp-usage-message
    |   `-- +zi-message
    |-- additional.zsh/.zi-clear-debug-report
```

```
|-- additional.zsh/.zi-debug-start
|-- additional.zsh/.zi-debug-stop
|-- additional.zsh/.zi-debug-unload
|-- autoload.zsh/.zi-analytics-menu
|-- autoload.zsh/.zi-cdisable
|-- autoload.zsh/.zi-cenable
|-- autoload.zsh/.zi-clear-completions
|-- autoload.zsh/.zi-compile-uncompile-all
|-- autoload.zsh/.zi-compiled
|-- autoload.zsh/.zi-help
|-- autoload.zsh/.zi-list-bindkeys
|-- autoload.zsh/.zi-list-compdef-replay
|-- autoload.zsh/.zi-ls
|-- autoload.zsh/.zi-module
|-- autoload.zsh/.zi-recently
|-- autoload.zsh/.zi-registered-ice-mods
|-- autoload.zsh/.zi-registered-subcommands
|-- autoload.zsh/.zi-search-completions
|-- autoload.zsh/.zi-self-update
|-- autoload.zsh/.zi-show-all-reports
|-- autoload.zsh/.zi-show-completions
|-- autoload.zsh/.zi-show-debug-report
|-- autoload.zsh/.zi-show-registered-plugins
|-- autoload.zsh/.zi-show-report
|-- autoload.zsh/.zi-show-times
|-- autoload.zsh/.zi-show-zstatus
|-- autoload.zsh/.zi-uncompile-plugin
|-- autoload.zsh/.zi-uninstall-completions
|-- autoload.zsh/.zi-unload
|-- autoload.zsh/.zi-update-or-status
|-- autoload.zsh/.zi-update-or-status-all
|-- compinit
|-- install.zsh/.zi-compile-plugin
|-- install.zsh/.zi-compinit
|-- install.zsh/.zi-forget-completion
`-- install.zsh/.zi-install-completions
```

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

zicdclear

```
]]]
FUNCTION: zicdclear. [[[
A wrapper for `zi cdclear -q' which can be called from hook ices like the
atinit'', atload'', etc. ices.
```

Has 1 line(s). Calls functions:

```
zicdclear
```

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

zicdreplay

```
FUNCTION: zicdreplay. [[[
A function that can be invoked from within `atinit', `atload', etc. ice-mod.
It works like `zi cdreplay', which cannot be invoked from such hook ices.
```

Has 1 line(s). Calls functions:

```
zicdreplay
```

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

zicompdef

```
]]]
FUNCTION: zicompdef. [[[
Stores compdef for a replay with `zicdreplay' (turbo mode) or with `zi cdreplay'
(normal mode).
An utility function of an undefined use case.
```

Has 1 line(s). Doesn't call other functions.

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

zicompinit

```
]]]
FUNCTION: zicompinit. [[[
A function that can be invoked from within `atinit', `atload', etc. ice-mod.
It runs `autoload compinit; compinit' and respects
ZI[ZCOMPDUMP_PATH] and ZI[COMPINIT_OPTS].
```

Has 1 line(s). Calls functions:

```
zicompinit
`-- compinit
```

Uses feature(s): *autoload, compinit*

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

zicompinit_fast

```
]]]
FUNCTION: zicompinit_fast. [[[
Checking the cached .zcompdump file to see if it must be regenerated adds a
noticable delay to zsh startup.
This restricts checking it once a day, determines when to regenerate, as compinit
doesn't always need to
modify the compdump and compiles mapped to share (total mem reduction) run in
background in multiple shells.
A function that can be invoked from within `atinit', `atload'
```

Has 13 line(s). Calls functions:

```
zicompinit_fast
`-- compinit
```

Uses feature(s): *autoload, compinit, zcompile*

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

zpcdclear

Has 1 line(s). Calls functions:

```
zpcdclear
```

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

zpcdreplay

Has 1 line(s). Calls functions:

```
zpcdreplay
```

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

zpcompdef

Has 1 line(s). Doesn't call other functions.

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

zpcompinit

Has 1 line(s). Calls functions:

```
zpcompinit
`-- compinit
```

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

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

▯▼▯ ˷

```
]]]
Compatibility functions. [[[
```

Has 1 line(s). Calls functions:

```
▯▯▯
`-- zi
    |-- +zi-message
    |-- +zi-prehelp-usage-message
    |   `-- +zi-message
    |-- additional.zsh/.zi-clear-debug-report
    |-- additional.zsh/.zi-debug-start
    |-- additional.zsh/.zi-debug-stop
    |-- additional.zsh/.zi-debug-unload
    |-- autoload.zsh/.zi-analytics-menu
    |-- autoload.zsh/.zi-cdisable
    |-- autoload.zsh/.zi-cenable
    |-- autoload.zsh/.zi-clear-completions
    |-- autoload.zsh/.zi-compile-uncompile-all
    |-- autoload.zsh/.zi-compiled
    |-- autoload.zsh/.zi-help
    |-- autoload.zsh/.zi-list-bindkeys
    |-- autoload.zsh/.zi-list-compdef-replay
    |-- autoload.zsh/.zi-ls
    |-- autoload.zsh/.zi-module
    |-- autoload.zsh/.zi-recently
    |-- autoload.zsh/.zi-registered-ice-mods
    |-- autoload.zsh/.zi-registered-subcommands
```

```
|-- autoload.zsh/.zi-search-completions
|-- autoload.zsh/.zi-self-update
|-- autoload.zsh/.zi-show-all-reports
|-- autoload.zsh/.zi-show-completions
|-- autoload.zsh/.zi-show-debug-report
|-- autoload.zsh/.zi-show-registered-plugins
|-- autoload.zsh/.zi-show-report
|-- autoload.zsh/.zi-show-times
|-- autoload.zsh/.zi-show-zstatus
|-- autoload.zsh/.zi-uncompile-plugin
|-- autoload.zsh/.zi-uninstall-completions
|-- autoload.zsh/.zi-unload
|-- autoload.zsh/.zi-update-or-status
|-- autoload.zsh/.zi-update-or-status-all
|-- compinit
|-- install.zsh/.zi-compile-plugin
|-- install.zsh/.zi-compinit
|-- install.zsh/.zi-forget-completion
`-- install.zsh/.zi-install-completions
```

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

add-zsh-hook

```
Add to HOOK the given FUNCTION.
HOOK is one of chpwd, precmd, preexec, periodic, zshaddhistory,
zshexit, zsh_directory_name (the _functions subscript is not required).
```

```
With -d, remove the function from the hook instead; delete the hook
variable if it is empty.
```

```
-D behaves like -d, but pattern characters are active in the
function name, so any matching function will be deleted from the hook.
```

Has 93 line(s). Doesn't call other functions.

Uses feature(s): *autoload*, *getopts*

Called by:

```
@zi-scheduler
Script-Body
```

compinit

> 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 549 line(s). Doesn't call other functions.

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

Called by:

```
zi
zicompinit
zicompinit_fast
zpcompinit
```

is-at-least

> Test whether $ZSH_VERSION (or some value of your choice, if a second argument
> is provided) is greater than or equal to x.y.z-r (in argument one). In fact,
> it'll accept any dot/dash-separated string of numbers as its second argument
> and compare it to the dot/dash-separated first argument. Leading non-number
> parts of a segment (such as the "zefram" in 3.1.2-zefram4) are not considered
> when the comparison is done; only the numbers matter. Any left-out segments
> in the first argument that are present in the version string compared are
> considered as zeroes, eg 3 == 3.0 == 3.0.0 == 3.0.0.0 and so on.

Has 56 line(s). Doesn't call other functions.

Called by:

```
:zi-tmp-subst-autoload
:zi-tmp-subst-bindkey
Script-Body
```