zinit-autoload.zsh(1)

NAME

zinit-autoload.zsh - a shell script

SYNOPSIS

Documentation automatically generated with `zshelldoc'

FUNCTIONS

```
.zinit-any-to-uspl2
.zinit-at-eval
.zinit-build-module
.zinit-cd
.zinit-cdisable
.zinit-cenable
.zinit-changes
.zinit-check-comp-consistency
.zinit-check-which-completions-are-enabled
.zinit-check-which-completions-are-installed
.zinit-clear-completions
.zinit-clear-report-for
.zinit-compiled
.zinit-compile-uncompile-all
.zinit-confirm
.zinit-create
.zinit-delete
.zinit-diff-env-compute
.zinit-diff-functions-compute
.zinit-diff-options-compute
.zinit-diff-parameter-compute
.zinit-edit
.zinit-exists-message
.zinit-find-completions-of-plugin
.zinit-format-env
.zinit-format-functions
.zinit-format-options
.zinit-format-parameter
.zinit-get-completion-owner
.zinit-get-completion-owner-uspl2col
.zinit-get-path
.zinit-glance
.zinit-help
.zinit-list-bindkeys
```

```
.zinit-list-compdef-replay
.zinit-ls
.zinit-module
.zinit-pager
.zinit-prepare-readlink
.zinit-recall
.zinit-recently
.zinit-restore-extendedglob
.zinit-run-delete-hooks
.zinit-save-set-extendedglob
.zinit-search-completions
.zinit-self-update
.zinit-show-all-reports
.zinit-show-completions
.zinit-show-debug-report
.zinit-show-registered-plugins
.zinit-show-report
.zinit-show-times
.zinit-show-zstatus
.zinit-stress
.zinit-uncompile-plugin
.zinit-uninstall-completions
.zinit-unload
.zinit-unregister-plugin
.zinit-update-all-parallel
.zinit-update-or-status
.zinit-update-or-status-all
.zinit-update-or-status-snippet
.zinit-wait-for-update-jobs
```

DETAILS

Script Body

Has 5 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*

zinit-any-to-uspl2

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

```
FUNCTION: .zinit-any-to-uspl2 [[[
Converts given plugin-spec to format that's used in keys for hash tables.
So basically, creates string "user/plugin" (this format is called: uspl2).
```

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

Has 2 line(s). Calls functions:

```
.zinit-any-to-uspl2
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Called by:

```
.zinit-clear-report-for
.zinit-exists-message
```

zinit-at-eval

~~~~~~

```
FUNCTION: .zinit-at-eval [[[
```

Has 5 line(s). Calls functions:

```
.zinit-at-eval
`-- zinit.zsh/@zinit-substitute
```

Uses feature(s): *eval*

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

zinit-build-module

~~~~~~~

```
]]]
FUNCTION: .zinit-build-module [[[
Performs ./configure && make on the module and displays information
how to load the module in .zshrc.
```

Has 29 line(s). Calls functions:

```
.zinit-build-module
`-- zinit.zsh/+zinit-message
```

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

Called by:

```
.zinit-module
```

zinit-cd

~~~

> ```
> FUNCTION: .zinit-cd [[[
> Jumps to plugin's directory (in Zinit's home directory).
> ```
>
> ```
> User-action entry point.
> ```
>
> ```
> $1 - plugin spec (4 formats: user---plugin, user/plugin, user, plugin)
> $2 - plugin (only when $1 - i.e. user - given)
> ```

Has 15 line(s). Calls functions:

```
.zinit-cd
`-- zinit.zsh/+zinit-message
```

Uses feature(s): *setopt*

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

zinit-cdisable

~~~~~

> ```
> FUNCTION: .zinit-cdisable [[[
> Enables given installed completion.
> ```
>
> ```
> User-action entry point.
> ```
>
> ```
> $1 - e.g. "_mkdir" or "mkdir"
> ```

Has 30 line(s). Calls functions:

```
.zinit-cdisable
```

Called by:

```
zinit.zsh/zinit
```

zinit-cenable

~~~~~~

```
FUNCTION: .zinit-cenable [[[
Disables given installed completion.
```

```
User-action entry point.
```

```
$1 - e.g. "_mkdir" or "mkdir"
```

Has 31 line(s). Calls functions:

```
.zinit-cenable
```

Called by:

```
zinit.zsh/zinit
```

zinit-changes

~~~~~~

```
]]]
FUNCTION: .zinit-changes [[[
Shows `git log` of given plugin.
```

```
User-action entry point.
```

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

Has 9 line(s). Calls functions:

```
.zinit-changes
|-- zinit-side.zsh/.zinit-exists-physically-message
`-- zinit.zsh/.zinit-any-to-user-plugin
```

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

zinit-check-comp-consistency

~~~~~~~~~~

```
FUNCTION: .zinit-check-comp-consistency [[[
Zinit creates symlink for each installed completion.
This function checks whether given completion (i.e.
file like "_mkdir") is indeed a symlink. Backup file
is a completion that is disabled - has the leading "_"
removed.
```

```
$1 - path to completion within plugin's directory
$2 - path to backup file within plugin's directory
```

Has 11 line(s). Doesn't call other functions.

Called by:

```
.zinit-cdisable
.zinit-cenable
```

zinit-check-which-completions-are-enabled

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

```
FUNCTION: .zinit-check-which-completions-are-enabled [[[
For each argument that each should be a path to completion
within a plugin's dir, it checks whether that completion
is disabled - returns 0 or 1 on corresponding positions
in reply.
```

```
Uninstalled completions will be reported as "0"
- i.e. disabled
```

```
$1, ... - path to completion within plugin's directory
```

Has 11 line(s). Doesn't call other functions.

Called by:

```
.zinit-show-report
```

zinit-check-which-completions-are-installed

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

> ```
> FUNCTION: .zinit-check-which-completions-are-installed [[[
> For each argument that each should be a path to completion
> within a plugin's dir, it checks whether that completion
> is installed - returns 0 or 1 on corresponding positions
> in reply.
> ```
>
> ```
> $1, ... - path to completion within plugin's directory
> ```

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

Called by:

```
.zinit-show-report
```

zinit-clear-completions

~~~~~~~~

> ```
> FUNCTION: .zinit-clear-completions [[[
> Delete stray and improper completions.
> ```
>
> ```
> Completions live even when plugin isn't loaded - if they are
> installed and enabled.
> ```
>
> ```
> User-action entry point.
> ```

Has 37 line(s). Calls functions:

```
.zinit-clear-completions
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/.zinit-prepare-home
zinit.zsh/zinit
```

zinit-clear-report-for

~~~~~~~~

> ```
> FUNCTION: .zinit-clear-report-for [[[
> Clears all report data for given user/plugin. This is
> done by resetting all related global ZINIT_* hashes.
> ```
>
> ```
> $1 - plugin spec (4 formats: user---plugin, user/plugin, user, plugin)
> $2 - (optional) plugin (only when $1 - i.e. user - given)
> ```

Has 23 line(s). Calls functions:

```
.zinit-clear-report-for
```

Called by:

```
.zinit-unload
```

zinit-compiled

~~~~~

> ```
> FUNCTION: .zinit-compiled [[[
> Displays list of plugins that are compiled.
> ```
>
> ```
> User-action entry point.
> ```

Has 26 line(s). Calls functions:

```
.zinit-compiled
|-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-compile-uncompile-all

~~~~~~~~~

> ```
> FUNCTION: .zinit-compile-uncompile-all [[[
> Compiles or uncompiles all existing (on disk) plugins.
> ```
>
> ```
> User-action entry point.
> ```

Has 23 line(s). Calls functions:

```
.zinit-compile-uncompile-all
|-- zinit-install.zsh/.zinit-compile-plugin
|-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-confirm

~~~~~~

> ```
> FUNCTION: .zinit-confirm [[[
> Prints given question, waits for "y" key, evals
> given expression if "y" obtained
> ```
>
> ```
> $1 - question
> $2 - expression
> ```

Has 22 line(s). Doesn't call other functions.

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

Called by:

```
.zinit-delete
```

zinit-create

~~~~~

```
FUNCTION: .zinit-create [[[
Creates a plugin, also on Github (if not "_local/name" plugin).
```

```
User-action entry point.
```

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

Has 109 line(s). Calls functions:

```
.zinit-create
|-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
|-- zinit-side.zsh/.zinit-exists-physically
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Uses feature(s): *autoload*, *setopt*, *vared*

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

zinit-delete

~~~~~

```
]]]
FUNCTION: .zinit-delete [[[
Deletes plugin's or snippet's directory (in Zinit's home directory).
```

```
User-action entry point.
```

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

Has 99 line(s). Calls functions:

```
.zinit-delete
|-- zinit-side.zsh/.zinit-compute-ice
|-- zinit.zsh/.zinit-any-to-user-plugin
|-- zinit.zsh/.zinit-parse-opts
`-- zinit.zsh/+zinit-prehelp-usage-message
```

Uses feature(s): *setopt*

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

zinit-diff-env-compute

~~~~~~~~

> ```
> FUNCTION: .zinit-diff-env-compute [[[
> Computes ZINIT_PATH, ZINIT_FPATH that hold (f)path components
> added by plugin. Uses data gathered earlier by .zinit-diff-env().
> ```
>
> ```
> $1 - user/plugin
> ```

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

Uses feature(s): *setopt*

Called by:

```
.zinit-show-report
.zinit-unload
```

zinit-diff-functions-compute

~~~~~~~~~~

> ```
> FUNCTION: .zinit-diff-functions-compute [[[
> Computes FUNCTIONS that holds new functions added by plugin.
> Uses data gathered earlier by .zinit-diff-functions().
> ```
>
> ```
> $1 - user/plugin
> ```

Has 19 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
.zinit-show-report
.zinit-unload
```

zinit-diff-options-compute

~~~~~~~~

> ```
> FUNCTION: .zinit-diff-options-compute [[[
> Computes OPTIONS that holds options changed by plugin.
> Uses data gathered earlier by .zinit-diff-options().
> ```
>
> ```
> $1 - user/plugin
> ```

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

Uses feature(s): *setopt*

Called by:

```
.zinit-show-report
.zinit-unload
```

zinit-diff-parameter-compute

~~~~~~~~~~

> ```
> FUNCTION: .zinit-diff-parameter-compute [[[
> Computes ZINIT_PARAMETERS_PRE, ZINIT_PARAMETERS_POST that hold
> parameters created or changed (their type) by plugin. Uses
> data gathered earlier by .zinit-diff-parameter().
> ```
>
> ```
> $1 - user/plugin
> ```

Has 28 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
.zinit-show-report
.zinit-unload
```

zinit-edit

~~~~
~~~~

```
FUNCTION: .zinit-edit [[[
Runs $EDITOR on source of given plugin. If the variable is not
set then defaults to `vim'.
```

```
User-action entry point.
```

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

Has 22 line(s). Calls functions:

```
.zinit-edit
`-- zinit-side.zsh/.zinit-compute-ice
```

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

zinit-exists-message

~~~~~~~

```
FUNCTION: .zinit-exists-message [[[
Checks if plugin is loaded. Testable. Also outputs error
message if plugin is not loaded.
```

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

Has 7 line(s). Calls functions:

```
.zinit-exists-message
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Called by:

```
.zinit-show-report
.zinit-unload
```

zinit-find-completions-of-plugin

~~~~~~~~~~~

```
FUNCTION: .zinit-find-completions-of-plugin [[[
Searches for completions owned by given plugin.
Returns them in `reply' array.
```

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

Has 6 line(s). Calls functions:

```
.zinit-find-completions-of-plugin
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
.zinit-show-report
```

zinit-format-env

~~~~~~~~

```
FUNCTION: .zinit-format-env [[[
Creates one-column text about FPATH or PATH elements
added when given plugin was loaded.
```

```
$1 - user/plugin (i.e. uspl2 format of plugin-spec)
$2 - if 1, then examine PATH, if 2, then examine FPATH
```

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

Called by:

```
.zinit-show-report
```

zinit-format-functions

~~~~~~~~

```
FUNCTION: .zinit-format-functions [[[
Creates a one or two columns text with functions created
by given plugin.
```

```
$1 - user/plugin (i.e. uspl2 format of plugin-spec)
```

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

Called by:

```
.zinit-show-report
```

zinit-format-options

~~~~~~~

```
FUNCTION: .zinit-format-options [[[
Creates one-column text about options that changed when
plugin "$1" was loaded.
```

```
$1 - user/plugin (i.e. uspl2 format of plugin-spec)
```

Has 21 line(s). Calls functions:

```
.zinit-format-options
```

Called by:

```
.zinit-show-report
```

zinit-format-parameter

~~~~~~~~~~

```
FUNCTION: .zinit-format-parameter [[[
Creates one column text that lists global parameters that
changed when the given plugin was loaded.
```

```
$1 - user/plugin (i.e. uspl2 format of plugin-spec)
```

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

Uses feature(s): *setopt*

Called by:

```
.zinit-show-report
```

zinit-get-completion-owner

~~~~~~~~~

```
FUNCTION: .zinit-get-completion-owner [[[
Returns "user---plugin" string (uspl1 format) of plugin that
owns given completion.
```

```
Both :A and readlink will be used, then readlink's output if
results differ. Readlink might not be available.
```

```
:A will read the link "twice" and give the final repository
directory, possibly without username in the uspl format;
readlink will read the link "once"
```

```
$1 - absolute path to completion file (in COMPLETIONS_DIR)
$2 - readlink command (":" or "readlink")
```

Has 22 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
.zinit-clear-completions
.zinit-get-completion-owner-uspl2col
.zinit-show-completions
```

zinit-get-completion-owner-uspl2col

~~~~~~~~~~~

```
FUNCTION: .zinit-get-completion-owner-uspl2col [[[
For shortening of code - returns colorized plugin name
that owns given completion.
```

```
$1 - absolute path to completion file (in COMPLETIONS_DIR)
$2 - readlink command (":" or "readlink")
```

Has 2 line(s). Calls functions:

```
.zinit-get-completion-owner-uspl2col
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Called by:

```
.zinit-cdisable
.zinit-cenable
```

zinit-get-path

~~~~~

```
]]]
FUNCTION: .zinit-get-path [[[
Returns path of given ID-string, which may be a plugin-spec
(like "user/plugin" or "user" "plugin"), an absolute path
("%" "/home/..." and also "%SNIPPETS/..." etc.), or a plugin
nickname (i.e. id-as'' ice-mod), or a snippet nickname.
```

Has 8 line(s). Calls functions:

```
.zinit-get-path
`-- zinit.zsh/.zinit-get-object-path
```

Uses feature(s): *setopt*

Called by:

```
.zinit-cd
.zinit-uninstall-completions
```

zinit-glance

~~~~~

```
FUNCTION: .zinit-glance [[[
Shows colorized source code of plugin. Is able to use pygmentize,
highlight, GNU source-highlight.
```

```
User-action entry point.
```

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

Has 39 line(s). Calls functions:

```
.zinit-glance
|-- zinit-side.zsh/.zinit-exists-physically-message
|-- zinit-side.zsh/.zinit-first
`-- zinit.zsh/.zinit-any-to-user-plugin
```

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

zinit-help

~~~~~~

```
FUNCTION: .zinit-help [[[
Shows usage information.
```

```
User-action entry point.
```

Has 67 line(s). Doesn't call other functions.

Called by:

```
zinit.zsh/zinit
```

zinit-list-bindkeys

~~~~~~~~

```
]]]
FUNCTION: .zinit-list-bindkeys [[[
```

Has 44 line(s). Calls functions:

```
.zinit-list-bindkeys
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Called by:

```
zinit.zsh/zinit
```

zinit-list-compdef-replay

~~~~~~~~~~

> ```
> FUNCTION: .zinit-list-compdef-replay [[[
> Shows recorded compdefs (called by plugins loaded earlier).
> Plugins often call `compdef' hoping for `compinit' being
> already ran. Zinit solves this by recording compdefs.
> ```
>
> ```
> User-action entry point.
> ```

Has 5 line(s). Doesn't call other functions.

Called by:

```
zinit.zsh/zinit
```

zinit-ls

~~~

> ```
> FUNCTION: .zinit-ls [[[
> ```

Has 19 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-module

~~~~~
~~~~~

```
]]]
FUNCTION: .zinit-module [[[
Function that has sub-commands passed as long-options (with two dashes, --).
It's an attempt to plugin only this one function into `zinit' function
defined in zinit.zsh, to not make this file longer than it's needed.
```

Has 24 line(s). Calls functions:

```
.zinit-module
```

Called by:

```
.zinit-build-module
zinit.zsh/Script-Body
zinit.zsh/zinit
```

zinit-pager

~~~~

```
FUNCTION: .zinit-pager [[[
BusyBox less lacks the -X and -i options, so it can use more
```

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

Uses feature(s): *setopt*

Called by:

```
.zinit-glance
.zinit-self-update
.zinit-update-or-status
```

zinit-prepare-readlink

~~~~~~~~

```
FUNCTION: .zinit-prepare-readlink [[[
Prepares readlink command, used for establishing completion's owner.
```

```
$REPLY = ":" or "readlink"
```

Has 4 line(s). Doesn't call other functions.

Uses feature(s): *type*

Called by:

```
.zinit-cdisable
.zinit-cenable
.zinit-clear-completions
.zinit-show-completions
```

zinit-recall

~~~~~

```
]]]
FUNCTION: .zinit-recall [[[
```

Has 38 line(s). Calls functions:

```
.zinit-recall
|-- zinit-side.zsh/.zinit-compute-ice
`-- zinit.zsh/+zinit-deploy-message
```

Uses feature(s): *setopt*

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

zinit-recently

~~~~~

```
FUNCTION: .zinit-recently [[[
Shows plugins that obtained commits in specified past time.
```

```
User-action entry point.
```

```
$1 - time spec, e.g. "1 week"
```

Has 28 line(s). Calls functions:

```
.zinit-recently
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-restore-extendedglob

~~~~~~~~

```
FUNCTION: .zinit-restore-extendedglob [[[
Restores extendedglob-option from state saved earlier.
```

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

Uses feature(s): *setopt*

Called by:

```
.zinit-format-options
.zinit-unload
```

zinit-run-delete-hooks

~~~~~~~~~~

```
FUNCTION: .zinit-run-delete-hooks [[[
```

Has 17 line(s). Calls functions:

```
.zinit-run-delete-hooks
`-- zinit-side.zsh/.zinit-countdown
```

Uses feature(s): *eval*

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

zinit-save-set-extendedglob

~~~~~~~~~

```
FUNCTION: .zinit-save-set-extendedglob [[[
Enables extendedglob-option first saving if it was already
enabled, for restoration of this state later.
```

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

Uses feature(s): *setopt*
~~~~~~~~~

Called by:

```
.zinit-format-options
.zinit-unload
```

zinit-search-completions

~~~~~~~~

> ```
> FUNCTION: .zinit-search-completions [[[
> While .zinit-show-completions() shows what completions are
> installed, this functions searches through all plugin dirs
> showing what's available in general (for installation).
> ```
>
> ```
> User-action entry point.
> ```

Has 43 line(s). Calls functions:

```
.zinit-search-completions
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-self-update

~~~~~~

> ```
> FUNCTION: .zinit-self-update [[[
> Updates Zinit code (does a git pull).
> ```
>
> ```
> User-action entry point.
> ```

Has 46 line(s). Calls functions:

```
.zinit-self-update
|-- zinit.zsh/.zinit-get-mtime-into
`-- zinit.zsh/+zinit-message
```

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

Called by:

```
.zinit-update-or-status-all
zinit.zsh/zinit
```

zinit-show-all-reports

~~~~~~~~

> ```
> FUNCTION: .zinit-show-all-reports [[[
> Displays reports of all loaded plugins.
> ```
>
> ```
> User-action entry point.
> ```

Has 5 line(s). Calls functions:

```
.zinit-show-all-reports
```

Called by:

```
zinit.zsh/zinit
```

zinit-show-completions

~~~~~~~~

> ```
> FUNCTION: .zinit-show-completions [[[
> Display installed (enabled and disabled), completions. Detect
> stray and improper ones.
> ```
>
> ```
> Completions live even when plugin isn't loaded - if they are
> installed and enabled.
> ```
>
> ```
> User-action entry point.
> ```

Has 72 line(s). Calls functions:

```
.zinit-show-completions
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-show-debug-report

~~~~~~~~

> ```
> FUNCTION: .zinit-show-debug-report [[[
> Displays dtrace report (data recorded in interactive session).
> ```
>
> ```
> User-action entry point.
> ```

Has 1 line(s). Calls functions:

```
.zinit-show-debug-report
```

Called by:

```
zinit.zsh/zinit
```

zinit-show-registered-plugins

~~~~~~~~~~

> ```
> FUNCTION: .zinit-show-registered-plugins [[[
> Lists loaded plugins (subcommands list, lodaded).
> ```
>
> ```
> User-action entry point.
> ```

Has 22 line(s). Calls functions:

```
.zinit-show-registered-plugins
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-show-report

~~~~~~

```
FUNCTION: .zinit-show-report [[[
Displays report of the plugin given.
```

```
User-action entry point.
```

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

Has 71 line(s). Calls functions:

```
.zinit-show-report
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
.zinit-show-all-reports
.zinit-show-debug-report
zinit.zsh/zinit
```

zinit-show-times

~~~~~~~~

```
FUNCTION: .zinit-show-times [[[
Shows loading times of all loaded plugins.
```

```
User-action entry point.
```

Has 60 line(s). Calls functions:

```
.zinit-show-times
`-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-show-zstatus

~~~~~~

```
]]]
FUNCTION: .zinit-show-zstatus [[[
Shows Zinit status, i.e. number of loaded plugins,
of available completions, etc.
```

```
User-action entry point.
```

Has 47 line(s). Calls functions:

```
.zinit-show-zstatus
`-- zinit.zsh/+zinit-message
```

Uses feature(s): *setopt*

Called by:

```
zinit.zsh/zinit
```

zinit-stress

~~~~~

```
FUNCTION: .zinit-stress [[[
Compiles plugin with various options on and off to see
how well the code is written. The options are:
```

```
NO_SHORT_LOOPS, IGNORE_BRACES, IGNORE_CLOSE_BRACES, SH_GLOB,
CSH_JUNKIE_QUOTES, NO_MULTI_FUNC_DEF.
```

```
User-action entry point.
```

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

Has 38 line(s). Calls functions:

```
.zinit-stress
|-- zinit-side.zsh/.zinit-exists-physically-message
|-- zinit-side.zsh/.zinit-first
`-- zinit.zsh/.zinit-any-to-user-plugin
```

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

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

zinit-uncompile-plugin

~~~~~~~~

> ```
> FUNCTION: .zinit-uncompile-plugin [[[
> Uncompiles given plugin.
> ```
>
> ```
> User-action entry point.
> ```
>
> ```
> $1 - plugin spec (4 formats: user---plugin, user/plugin, user (+ plugin in $2),
> plugin)
> $2 - plugin (only when $1 - i.e. user - given)
> ```

Has 22 line(s). Calls functions:

```
.zinit-uncompile-plugin
|-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
.zinit-compile-uncompile-all
zinit.zsh/zinit
```

zinit-uninstall-completions

~~~~~~~~~

```
FUNCTION: .zinit-uninstall-completions [[[
Removes all completions of given plugin from Zshell (i.e. from FPATH).
The FPATH is typically `~/.zinit/completions/'.
```

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

Has 46 line(s). Calls functions:

```
.zinit-uninstall-completions
|-- zinit-install.zsh/.zinit-compinit
|-- zinit-install.zsh/.zinit-forget-completion
`-- zinit.zsh/+zinit-message
```

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

Called by:

```
zinit.zsh/zinit
```

zinit-unload

~~~~~

```
FUNCTION: .zinit-unload [[[
0. Call the Zsh Plugin's Standard *_plugin_unload function
0. Call the code provided by the Zsh Plugin's Standard @zsh-plugin-run-at-update
1. Delete bindkeys (...)
2. Delete Zstyles
3. Restore options
4. Remove aliases
5. Restore Zle state
6. Unfunction functions (created by plugin)
7. Clean-up FPATH and PATH
8. Delete created variables
9. Forget the plugin
```

```
User-action entry point.
```

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

Has 394 line(s). Calls functions:
~~~~~

```
.zinit-unload
|-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Uses feature(s): *alias*, *bindkey*, *eval*, *setopt*, *unalias*, *unfunction*, *zle*, *zstyle*

Called by:

```
zinit.zsh/.zinit-run-task
zinit.zsh/zinit
```

zinit-unregister-plugin

~~~~~~~~

> ```
> FUNCTION: .zinit-unregister-plugin [[[
> Removes the plugin from ZINIT_REGISTERED_PLUGINS array and from the
> zsh_loaded_plugins array (managed according to the plugin standard)
> ```

Has 6 line(s). Calls functions:

```
.zinit-unregister-plugin
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Called by:

```
.zinit-unload
```

zinit-update-all-parallel

~~~~~~~~~~

> ```
> FUNCTION: .zinit-update-in-parallel [[[
> ```

Has 84 line(s). Calls functions:

```
.zinit-update-all-parallel
|-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
|-- zinit.zsh/.zinit-any-to-user-plugin
`-- zinit.zsh/+zinit-message
```

Uses feature(s): *setopt*

Called by:

```
.zinit-update-or-status-all
```

zinit-update-or-status

~~~~~~~~

```
FUNCTION: .zinit-update-or-status [[[
Updates (git pull) or does `git status' for given plugin.
```

```
User-action entry point.
```

```
$1 - "status" for status, other for update
$2 - plugin spec (4 formats: user---plugin, user/plugin, user (+ plugin in $2),
plugin)
$3 - plugin (only when $1 - i.e. user - given)
```

Has 300 line(s). Calls functions:

```
.zinit-update-or-status
|-- zinit-install.zsh/.zinit-get-latest-gh-r-url-part
|-- zinit-install.zsh/.zinit-setup-plugin-dir
|-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
|-- zinit-side.zsh/.zinit-compute-ice
|-- zinit-side.zsh/.zinit-exists-physically
|-- zinit-side.zsh/.zinit-exists-physically-message
|-- zinit-side.zsh/.zinit-store-ices
|-- zinit-side.zsh/.zinit-two-paths
|-- zinit.zsh/.zinit-any-to-user-plugin
|-- zinit.zsh/+zinit-message
`-- zinit.zsh/.zinit-set-m-func
```

Uses feature(s): *kill*, *read*, *setopt*, *source*, *trap*, *wait*

Called by:

```
.zinit-update-all-parallel
.zinit-update-or-status-all
zinit.zsh/zinit
```

zinit-update-or-status-all

~~~~~~~~~

```
]]]
FUNCTION: .zinit-update-or-status-all [[[
Updates (git pull) or does `git status` for all existing plugins.
This includes also plugins that are not loaded into Zsh (but exist
on disk). Also updates (i.e. redownloads) snippets.
```

```
User-action entry point.
```

Has 124 line(s). Calls functions:

```
.zinit-update-or-status-all
|-- zinit-install.zsh/.zinit-compinit
|-- zinit-side.zsh/.zinit-any-colorify-as-uspl2
|-- zinit.zsh/.zinit-any-to-user-plugin
|-- zinit.zsh/.zinit-get-mtime-into
`-- zinit.zsh/+zinit-message
```

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

Called by:

```
zinit.zsh/zinit
```

zinit-update-or-status-snippet

~~~~~~~~~~

```
FUNCTION: .zinit-update-or-status-snippet [[[
```

```
Implements update or status operation for snippet given by URL.
```

```
$1 - "status" or "update"
$2 - snippet URL
```

Has 34 line(s). Calls functions:

```
.zinit-update-or-status-snippet
|-- zinit-install.zsh/.zinit-update-snippet
`-- zinit-side.zsh/.zinit-compute-ice
```

Uses feature(s): *source*
~~~~~~~~~~

Called by:

```
.zinit-update-all-parallel
.zinit-update-or-status-all
.zinit-update-or-status
```

zinit-wait-for-update-jobs

~~~~~~~~

> ```
> ]]]
> FUNCTION: .zinit-wait-for-update-jobs [[[
> ```

Has 18 line(s). Calls functions:

```
.zinit-wait-for-update-jobs
`-- zinit.zsh/+zinit-message
```

Uses feature(s): *wait*

Called by:

```
.zinit-update-all-parallel
```
~~~~~~~~