

zinit-side.zsh(1)

NAME

zinit-side.zsh - a shell script

SYNOPSIS

Documentation automatically generated with `zshelldoc`

FUNCTIONS

```
.zinit-any-colorify-as-uspl2
.zinit-compute-ice
.zinit-countdown
.zinit-exists-physically
.zinit-exists-physically-message
.zinit-first
.zinit-get-plg-dir
.zinit-shands-exp
.zinit-store-ices
.zinit-two-paths
AUTOLOAD zmv
```

DETAILS

Script Body

Has 1 line(s). No functions are called (may set up e.g. a hook, a Zle widget bound to a key, etc.).

zinit-any-colorify-as-uspl2

~~~~~  
~

FUNCTION: .zinit-any-colorify-as-uspl2 []  
Returns ANSI-colorified "user/plugin" string, from any supported plugin spec (user---plugin, user/plugin, user plugin, plugin).

\$1 - plugin spec (4 formats: user---plugin, user/plugin, user, plugin)  
\$2 - plugin (only when \$1 - i.e. user - given)  
\$REPLY = ANSI-colorified "user/plugin" string

Has 11 line(s). Calls functions:

```
.zinit-any-colorify-as-uspl2  
'-- zinit.zsh/.zinit-any-to-user-plugin
```

Called by:

```
.zinit-exists-physically-message  
zinit-autoload.zsh/.zinit-clear-completions  
zinit-autoload.zsh/.zinit-compiled  
zinit-autoload.zsh/.zinit-compile-uncompile-all  
zinit-autoload.zsh/.zinit-create  
zinit-autoload.zsh/.zinit-exists-message  
zinit-autoload.zsh/.zinit-get-completion-owner-uspl2col  
zinit-autoload.zsh/.zinit-list-bindkeys  
zinit-autoload.zsh/.zinit-recently  
zinit-autoload.zsh/.zinit-search-completions  
zinit-autoload.zsh/.zinit-show-completions  
zinit-autoload.zsh/.zinit-show-registered-plugins  
zinit-autoload.zsh/.zinit-show-times  
zinit-autoload.zsh/.zinit-uncompile-plugin  
zinit-autoload.zsh/.zinit-unload  
zinit-autoload.zsh/.zinit-update-all-parallel  
zinit-autoload.zsh/.zinit-update-or-status-all  
zinit-autoload.zsh/.zinit-update-or-status  
zinit-install.zsh/.zinit-install-completions  
zinit-install.zsh/.zinit-setup-plugin-dir
```

*zinit-compute-ice*

#####

```
]]]
```

```
FUNCTION: .zinit-compute-ice [[[
```

Computes ZINIT\_ICE array (default, it can be specified via \$3) from a) input ZINIT\_ICE, b) static ice, c) saved ice, taking priorities into account. Also returns path to snippet directory and optional name of snippet file (only valid if ZINIT\_ICE[svn] is not set).

Can also pack resulting ices into ZINIT\_SICE (see \$2).

\$1 - URL (also plugin-spec)

\$2 - "pack" or "nopack" or "pack-nf" - packing means ZINIT\_ICE wins with static ice; "pack-nf" means that disk-ices will be ignored (no-file?)

\$3 - name of output associative array, "ZINIT\_ICE" is the default

\$4 - name of output string parameter, to hold path to directory ("local\_dir")

\$5 - name of output string parameter, to hold filename ("filename")

\$6 - name of output string parameter, to hold is-snippet 0/1-bool ("is\_snippet")

Has 111 line(s). Calls functions:

```
.zinit-compute-ice
|-- zinit.zsh/.zinit-any-to-user-plugin
|-- zinit.zsh/.zinit-pack-ice
`-- zmv
```

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

Called by:

```
zinit-autoload.zsh/.zinit-delete
zinit-autoload.zsh/.zinit-edit
zinit-autoload.zsh/.zinit-recall
zinit-autoload.zsh/.zinit-update-or-status-snippet
zinit-autoload.zsh/.zinit-update-or-status
zinit-install.zsh/.zinit-compile-plugin
```

*zinit-countdown*

~~~~~

```
]]]
```

```
FUNCTION: .zinit-countdown [[[
```

Displays a countdown 5...4... etc. and returns 0 if it successfully reaches 0, or 1 if Ctrl-C will be pressed.

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

Uses feature(s): *trap*

Called by:

```
zinit-autoload.zsh/.zinit-run-delete-hooks
zinit-autoload.zsh/.zinit-update-or-status
zinit-install.zsh/.zinit-download-snippet
zinit-install.zsh/.zinit-setup-plugin-dir
```

zinit-exists-physically

~~~~~

```
]]]
FUNCTION: .zinit-exists-physically [[[
Checks if directory of given plugin exists in PLUGIN_DIR.
```

Testable.

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

Has 10 line(s). Calls functions:

```
.zinit-exists-physically
`-- zinit.zsh/.zinit-any-to-user-plugin
```

Called by:

```
.zinit-exists-physically-message
zinit-autoload.zsh/.zinit-create
zinit-autoload.zsh/.zinit-get-path
zinit-autoload.zsh/.zinit-update-or-status
```

*zinit-exists-physically-message*

~~~~~

```
FUNCTION: .zinit-exists-physically-message [[[
Checks if directory of given plugin exists in PLUGIN_DIR,
and outputs error message if it doesn't.
```

Testable.

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

Has 11 line(s). Calls functions:

```
.zinit-exists-physically-message
```

Called by:

```
.zinit-compute-ice
zinit-autoload.zsh/.zinit-changes
zinit-autoload.zsh/.zinit-glance
zinit-autoload.zsh/.zinit-stress
zinit-autoload.zsh/.zinit-update-or-status
zinit-install.zsh/.zinit-install-completions
```

zinit-first

```
]]]
FUNCTION: .zinit-first [[[
Finds the main file of plugin. There are multiple file name
formats, they are ordered in order starting from more correct
ones, and matched. .zinit-load-plugin() has similar code parts
and doesn't call .zinit-first() for performance. Obscure matching
is done in .zinit-find-other-matches, here and in .zinit-load().
Obscure = non-standard main-file naming convention.
```

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

Has 17 line(s). Calls functions:

```
.zinit-first
|-- zinit.zsh/.zinit-any-to-user-plugin
`-- zinit.zsh/.zinit-find-other-matches
```

Called by:

```
zinit-autoload.zsh/.zinit-glance
zinit-autoload.zsh/.zinit-stress
zinit-install.zsh/.zinit-compile-plugin
```

zinit-get-plg-dir

~~~~~

```
FUNCTION: .zinit-plgdir [[
```

Has 10 line(s). Doesn't call other functions.

Called by:

```
.zinit-first
```

### *zinit-shands-exp*

~~~~~

```
FUNCTION: .zinit-shands-exp [[
Does expansion of currently little unstandarized
shorthands like "%SNIPPETS", "%HOME", "OMZ::", "PZT::".
```

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

Called by:

```
.zinit-compute-ice
.zinit-exists-physically-message
.zinit-exists-physically
zinit-autoload.zsh/.zinit-get-path
```

zinit-store-ices

~~~~~

```
]]]
FUNCTION: .zinit-store-ices [[[
Saves ice mods in given hash onto disk.
```

```
$1 - directory where to create / delete files
$2 - name of hash that holds values
$3 - additional keys of hash to store, space separated
$4 - additional keys of hash to store, empty-meaningful ices, space separated
$5 - the URL, if applicable
$6 - the mode (1 - svn, 0 - single file), if applicable
```

Has 32 line(s). Doesn't call other functions.

Called by:

```
zinit-autoload.zsh/.zinit-update-or-status
zinit-install.zsh/.zinit-download-snippet
zinit-install.zsh/.zinit-setup-plugin-dir
```

*zinit-two-paths*

~~~~~

```
FUNCTION: .zinit-two-paths [[[
Obtains a snippet URL without specification if it is an SVN URL (points to
directory) or regular URL (points to file), returns 2 possible paths for
further examination
```

Has 18 line(s). Calls functions:

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

Uses feature(s): *setopt*

Called by:

```
.zinit-compute-ice
zinit-autoload.zsh/.zinit-get-path
zinit-autoload.zsh/.zinit-update-or-status
```

zmv

```
function zmv {  
  zmv, zcp, zln:
```

This is a multiple move based on zsh pattern matching. To get the full power of it, you need a postgraduate degree in zsh. However, simple tasks work OK, so if that's all you need, here are some basic examples:

```
zmv '(*).txt' '$1.lis'
```

Rename foo.txt to foo.lis, etc. The parenthesis is the thing that gets replaced by the \$1 (not the '*', as happens in mmv, and note the '\$', not '=', so that you need to quote both words).

Has 299 line(s). Doesn't call other functions.

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

Called by:

```
.zinit-compute-ice
```