

side.zsh(1)

NAME

side.zsh - a shell script

SYNOPSIS

Documentation automatically generated with `zsdoc'

FUNCTIONS

```
Ė. zi-any-colorify-as-uspl2
Ė. zi-compute-ice
Ė. zi-countdown
Ė. zi-exists-physically
Ė. zi-exists-physically-message
Ė. zi-first
Ė. zi-store-ices
Ė. zi-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.).

zi-any-colorify-as-uspl2

~~~~~

FUNCTION: .zi-any-colorify-as-uspl2 [[[  
Returns ANSI-colored "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-colored "user/plugin" string

Has 22 line(s). Calls functions:

```
. zi -any-col ori fy-as-uspl 2  
|-- zi . zsh/. zi -any-to-pi d  
`-- zi . zsh/. zi -any-to-user-pl ugi n
```

Called by:

```
. zi -exi sts-physi cal ly-message  
autol oad. zsh/. zi -cl ear-compl eti ons  
autol oad. zsh/. zi -compi l e-uncompi l e-al l  
autol oad. zsh/. zi -compi l ed  
autol oad. zsh/. zi -create  
autol oad. zsh/. zi -exi sts-message  
autol oad. zsh/. zi -get-compl eti on-owner-uspl 2col  
autol oad. zsh/. zi -l i st-bi ndkeys  
autol oad. zsh/. zi -recentl y  
autol oad. zsh/. zi -search-compl eti ons  
autol oad. zsh/. zi -show-compl eti ons  
autol oad. zsh/. zi -show-regi stered-pl ugi ns  
autol oad. zsh/. zi -show-ti mes  
autol oad. zsh/. zi -uncompi l e-pl ugi n  
autol oad. zsh/. zi -unl oad  
autol oad. zsh/. zi -update-al l -paral l el  
autol oad. zsh/. zi -update-or-status-al l  
autol oad. zsh/. zi -update-or-status  
i nstal l . zsh/. zi -i nstal l -compl eti ons  
i nstal l . zsh/. zi -setup-pl ugi n-di r  
zi . zsh/. zi -formatter-pi d
```

*zi-compute-ice*

~~~~~

FUNCTION: `.zi-compute-ice` `[[[`
Computes ICE array (default, it can be specified via \$3) from a) input 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 ICE[svn] is not set).

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

\$1 - URL (also plugin-spec)
\$2 - "pack" or "nopack" or "pack-nf" - packing means ICE wins with static ice; "pack-nf" means that disk-ices will be ignored (no-file?)
\$3 - name of output associative array, "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 116 line(s). Calls functions:

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

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

Called by:

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

zi-countdown

~~~~~

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

Has 14 line(s). Calls functions:

```
.zi -countdown  
`-- zi .zsh/+zi -message
```

Uses feature(s): *trap*

Called by:

```
autoload.zsh/.zi -run-delete-hooks  
install.zsh/°zi -atclone-hook  
install.zsh/°zi -atpull-e-hook  
install.zsh/°zi -atpull-hook  
install.zsh/°zi -make-e-hook  
install.zsh/°zi -make-ee-hook  
install.zsh/°zi -make-hook
```

*zi-exists-physically*

~~~~~

```
FUNCTION: .zi-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 6 line(s). Calls functions:

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

Called by:

```
.zi-exists-physically-message  
autoload.zsh/.zi -create  
autoload.zsh/.zi -update-or-status
```

zi-exists-physically-message

~~~~~

FUNCTION: `.zi-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 22 line(s). Calls functions:

```
.zi-exists-physically-message  
|-- zi.zsh/+zi-message  
|-- zi.zsh/.zi-any-to-pid  
`-- zi.zsh/.zi-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

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

*zi-first*

~~~

FUNCTION: `.zi-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. `.zi-load-plugin()` has similar code parts and doesn't call `.zi-first()` for performance. Obscure matching is done in `.zi-find-other-matches`, here and in `.zi-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 19 line(s). Calls functions:

```
.zi -first
|-- zi.zsh/.zi -any-to-pid
|-- zi.zsh/.zi -any-to-user-plugin
|-- zi.zsh/.zi -find-other-matches
`-- zi.zsh/.zi -get-object-path
```

Called by:

```
.zi -two-paths
autoload.zsh/.zi -glance
autoload.zsh/.zi -stress
install.zsh/.zi -compile-plugin
```

zi-store-ices

~~~~~

```
FUNCTION: .zi-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 28 line(s). Doesn't call other functions.

Called by:

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

*zi-two-paths*

~~~~~

```
FUNCTION: .zi-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 22 line(s). Calls functions:

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

Uses feature(s): *setopt*

Called by:

```
.zi -compute-ice  
autoload.zsh/.zi -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:

```
.zi -compute-ice
```