

zplugin-autoload.zsh(1)

NAME

zplugin-autoload.zsh - a shell script

SYNOPSIS

Documentation automatically generated with `zshelldoc`

FUNCTIONS

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

```
-zplg-help
-zplg-list-bindkeys
-zplg-list-compdef-replay
-zplg-ls
-zplg-module
-zplg-prepare-readlink
-zplg-recall
-zplg-recently
-zplg-restore-extendedglob
-zplg-save-set-extendedglob
-zplg-search-completions
-zplg-self-update
-zplg-show-all-reports
-zplg-show-completions
-zplg-show-debug-report
-zplg-show-registered-plugins
-zplg-show-report
-zplg-show-times
-zplg-show-zstatus
-zplg-stress
-zplg-uncompile-plugin
-zplg-uninstall-completions
-zplg-unload
-zplg-update-or-status
-zplg-update-or-status-all
-zplg-update-or-status-snippet
AUTOLOAD compinit
```

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*

-zplg-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:

```
-zplg-any-to-uspl2  
'-- zplugin.zsh/-zplg-any-to-user-plugin
```

Called by:

```
-zplg-clear-report-for  
-zplg-exists-message
```

-zplg-at-eval

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

Uses feature(s): *eval*

Called by:

```
-zplg-update-or-status
```

-zplg-build-module

Performs `./configure && make` on the module and displays information how to load the module in `.zshrc`.

Has 27 line(s). Calls functions:

```
-zplg-build-module  
'-- -zplg-module
```

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

Called by:

```
-zplg-module
```

-zplg-cd

Jumps to plugin's directory (in Zplugin'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 13 line(s). Calls functions:

```
-zplg-cd
`-- -zplg-get-path
    |-- zplugin-side.zsh/-zplg-exists-physically
    |-- zplugin-side.zsh/-zplg-shands-exp
    |-- zplugin-side.zsh/-zplg-two-paths
    `-- zplugin.zsh/-zplg-any-to-user-plugin
```

Uses feature(s): *setopt*

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

-zplg-cdisable

Enables given installed completion.

User-action entry point.

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

Has 30 line(s). Calls functions:

```
-zplg-cdisable
|-- -zplg-check-comp-consistency
|-- -zplg-get-completion-owner-uspl2col
|   |-- -zplg-get-completion-owner
|   `-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
`-- -zplg-prepare-readlink
```

Called by:

zplugin.zsh/zplugin

-zplg-cenable

Disables given installed completion.

User-action entry point.

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

Has 31 line(s). Calls functions:

```
-zplg-cenable
|-- -zplg-check-comp-consistency
|-- -zplg-get-completion-owner-uspl2col
|   |-- -zplg-get-completion-owner
|   `-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
`-- -zplg-prepare-readlink
```

Called by:

zplugin.zsh/zplugin

-zplg-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:

```
-zplg-changes  
|-- zplugin-side.zsh/-zplg-exists-physically-message  
'-- zplugin.zsh/-zplg-any-to-user-plugin
```

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

-zplg-check-comp-consistency

Zplugin 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:

```
-zplg-cdisable  
-zplg-cenable
```

-zplg-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:

-zplg-show-report

-zplg-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:

-zplg-show-report

-zplg-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:

```
-zplg-clear-completions
|-- -zplg-get-completion-owner
|-- -zplg-prepare-readlink
`-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

zplugin.zsh/zplugin

-zplg-clear-report-for

Clears all report data for given user/plugin. This is done by resetting all related `ZPLG_*` 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:

```
-zplg-clear-report-for  
'-- -zplg-any-to-uspl2  
  '-- zplugin.zsh/-zplg-any-to-user-plugin
```

Called by:

```
-zplg-unload  
zplugin.zsh/-zplg-clear-debug-report
```

-zplg-compiled

Displays list of plugins that are compiled.

User-action entry point.

Has 26 line(s). Calls functions:

```
-zplg-compiled  
|-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2  
'-- zplugin.zsh/-zplg-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
zplugin.zsh/zplugin
```


-zplg-compile-uncompile-all

Compiles or uncompiles all existing (on disk) plugins.

User-action entry point.

Has 23 line(s). Calls functions:

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

Uses feature(s): *setopt*

Called by:

zplugin.zsh/zplugin

-zplg-compinit

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

No arguments.

Has 23 line(s). Calls functions:

```
-zplg-compinit
|-- compinit
`-- zplugin-install.zsh/-zplg-forget-completion
```

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

Called by:

zplugin.zsh/zplugin

-zplg-compute-ice

Computes ZPLG_ICE array (default, it can be specified via \$3) from a) input ZPLG_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 ZPLG_ICE[svn] is not set).

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

\$1 - URL (also plugin-spec)
\$2 - "pack" or "nopack" or "pack-nf" - packing means ZPLG_ICE wins with static ice;
"pack-nf" means that disk-ices will be ignored (no-file?)
\$3 - name of output associative array, "ZPLG_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")

Has 98 line(s). Calls functions:

```
-zplg-compute-ice  
|-- zplugin-side.zsh/-zplg-exists-physically-message  
|-- zplugin-side.zsh/-zplg-shands-exp  
|-- zplugin-side.zsh/-zplg-two-paths  
|-- zplugin.zsh/-zplg-any-to-user-plugin  
'-- zplugin.zsh/-zplg-pack-ice
```

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

Called by:

```
-zplg-recall  
-zplg-update-or-status-snippet  
-zplg-update-or-status
```

-zplg-confirm

Prints given question, waits for "y" key, evals given expression if "y" obtained

\$1 - question
\$2 - expression

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

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

Called by:

-zplg-delete

-zplg-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 67 line(s). Calls functions:

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

Uses feature(s): *vared*

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

-zplg-delete

Deletes plugin's or snippet's directory (in Zplugin'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 54 line(s). Calls functions:

```
-zplg-delete  
|-- -zplg-confirm  
|-- zplugin-side.zsh/-zplg-exists-physically-message  
|-- zplugin-side.zsh/-zplg-shands-exp  
|-- zplugin-side.zsh/-zplg-two-paths  
'-- zplugin.zsh/-zplg-any-to-user-plugin
```

Uses feature(s): *setopt*

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

-zplg-diff-env-compute

Computes ZPLG_PATH, ZPLG_FPATH that hold (f)path components added by plugin. Uses data gathered earlier by -zplg-diff-env().

\$1 - user/plugin

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

Uses feature(s): *setopt*

Called by:

```
-zplg-show-report  
-zplg-unload
```

-zplg-diff-functions-compute

Computes FUNCTIONS that holds new functions added by plugin.
Uses data gathered earlier by `-zplg-diff-functions()`.

\$1 - user/plugin

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

Uses feature(s): *setopt*

Called by:

`-zplg-show-report`
`-zplg-unload`

-zplg-diff-options-compute

Computes OPTIONS that holds options changed by plugin.
Uses data gathered earlier by `-zplg-diff-options()`.

\$1 - user/plugin

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

Uses feature(s): *setopt*

Called by:

`-zplg-show-report`
`-zplg-unload`

-zplg-diff-parameter-compute

Computes ZPLG_PARAMETERS_PRE, ZPLG_PARAMETERS_POST that hold parameters created or changed (their type) by plugin. Uses data gathered earlier by `-zplg-diff-parameter()`.

\$1 - user/plugin

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

Uses feature(s): *setopt*

Called by:

```
-zplg-show-report  
-zplg-unload
```

-zplg-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 13 line(s). Calls functions:

```
-zplg-edit  
|-- zplugin-side.zsh/-zplg-exists-physically-message  
|-- zplugin-side.zsh/-zplg-first  
'-- zplugin.zsh/-zplg-any-to-user-plugin
```

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

-zplg-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:

```
-zplg-exists-message  
|-- -zplg-any-to-uspl2  
|   '-- zplugin.zsh/-zplg-any-to-user-plugin  
|-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
```

Called by:

```
-zplg-show-report  
-zplg-unload
```

-zplg-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:

```
-zplg-find-completions-of-plugin  
 '-- zplugin.zsh/-zplg-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
-zplg-show-report
```

-zplg-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:

```
-zplg-show-report
```

-zplg-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:

```
-zplg-show-report
```

-zplg-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:

```
-zplg-format-options  
|-- -zplg-restore-extendedglob  
'-- -zplg-save-set-extendedglob
```

Called by:

```
-zplg-show-report
```

-zplg-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 34 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

-zplg-show-report

-zplg-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:

-zplg-clear-completions
-zplg-get-completion-owner-uspl2col
-zplg-show-completions

-zplg-get-completion-owner-uspl2col

For shortening of code - returns colored 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:

```
-zplg-get-completion-owner-uspl2col  
|-- -zplg-get-completion-owner  
'-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
```

Called by:

```
-zplg-cdisable  
-zplg-cenable
```

-zplg-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 35 line(s). Calls functions:

```
-zplg-get-path  
|-- zplugin-side.zsh/-zplg-exists-physically  
|-- zplugin-side.zsh/-zplg-shands-exp  
|-- zplugin-side.zsh/-zplg-two-paths  
'-- zplugin.zsh/-zplg-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
-zplg-cd  
-zplg-uninstall-completions
```

-zplg-glance

Shows colored 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:

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

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

-zplg-help

Shows usage information.

User-action entry point.

Has 68 line(s). Doesn't call other functions.

Called by:

```
zplugin.zsh/zplugin
```

-zplg-list-bindkeys

Has 42 line(s). Calls functions:

```
-zplg-list-bindkeys  
'-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
```

Called by:

zplugin.zsh/zplugin

-zplg-list-compdef-replay

Shows recorded compdefs (called by plugins loaded earlier).
Plugins often call 'compdef' hoping for 'compinit' being
already ran. Zplugin solves this by recording compdefs.

User-action entry point.

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

Called by:

zplugin.zsh/zplugin

-zplg-ls

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

Uses feature(s): *setopt*

Called by:

zplugin.zsh/zplugin

-zplg-module

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

Has 24 line(s). Calls functions:

-zplg-module
'-- -zplg-build-module

Called by:

```
-zplg-build-module  
zplugin.zsh/zplugin
```

-zplg-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:

```
-zplg-cdisable  
-zplg-cenable  
-zplg-clear-completions  
-zplg-show-completions
```

-zplg-recall

Has 37 line(s). Calls functions:

```
-zplg-recall  
'-- -zplg-compute-ice  
  |-- zplugin-side.zsh/-zplg-exists-physically-message  
  |-- zplugin-side.zsh/-zplg-shands-exp  
  |-- zplugin-side.zsh/-zplg-two-paths  
  |-- zplugin.zsh/-zplg-any-to-user-plugin  
  '-- zplugin.zsh/-zplg-pack-ice
```

Uses feature(s): *wait*

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

-zplg-recently

Shows plugins that obtained commits in specified past time.

User-action entry point.

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

Has 26 line(s). Calls functions:

```
-zplg-recently  
'-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

```
zplugin.zsh/zplugin
```

-zplg-restore-extendedglob

Restores extendedglob-option from state saved earlier.

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

Uses feature(s): *setopt*

Called by:

```
-zplg-format-options  
-zplg-show-registered-plugins  
-zplg-unload
```

-zplg-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:

```
-zplg-format-options  
-zplg-show-registered-plugins  
-zplg-unload
```

-zplg-search-completions

While `-zplg-show-completions()` shows what completions are installed, this function searches through all plugin dirs showing what's available in general (for installation).

User-action entry point.

Has 43 line(s). Calls functions:

```
-zplg-search-completions  
'-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

```
zplugin.zsh/zplugin
```

-zplg-self-update

Updates Zplugin code (does a git pull).

User-action entry point.

Has 23 line(s). Doesn't call other functions.

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

Called by:

```
zplugin.zsh/zplugin
```

-zplg-show-all-reports

Displays reports of all loaded plugins.

User-action entry point.

Has 5 line(s). Calls functions:

```
-zplg-show-all-reports
`-- -zplg-show-report
    |-- -zplg-check-which-completions-are-enabled
    |-- -zplg-check-which-completions-are-installed
    |-- -zplg-diff-env-compute
    |-- -zplg-diff-functions-compute
    |-- -zplg-diff-options-compute
    |-- -zplg-diff-parameter-compute
    |-- -zplg-exists-message
    |   |-- -zplg-any-to-uspl2
    |   |   `-- zplugin.zsh/-zplg-any-to-user-plugin
    |   `-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
    |-- -zplg-find-completions-of-plugin
    |   `-- zplugin.zsh/-zplg-any-to-user-plugin
    |-- -zplg-format-env
    |-- -zplg-format-functions
    |-- -zplg-format-options
    |   |-- -zplg-restore-extendedglob
    |   `-- -zplg-save-set-extendedglob
    |-- -zplg-format-parameter
    `-- zplugin.zsh/-zplg-any-to-user-plugin
```

Called by:

zplugin.zsh/zplugin

-zplg-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:

```
-zplg-show-completions  
|-- -zplg-get-completion-owner  
|-- -zplg-prepare-readlink  
'-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

zplugin.zsh/zplugin

-zplg-show-debug-report

Displays dtrace report (data recorded in interactive session).

User-action entry point.

Has 1 line(s). Calls functions:

```

-zplg-show-debug-report
`-- -zplg-show-report
    |-- -zplg-check-which-completions-are-enabled
    |-- -zplg-check-which-completions-are-installed
    |-- -zplg-diff-env-compute
    |-- -zplg-diff-functions-compute
    |-- -zplg-diff-options-compute
    |-- -zplg-diff-parameter-compute
    |-- -zplg-exists-message
    |   |-- -zplg-any-to-uspl2
    |   |   `-- zplugin.zsh/-zplg-any-to-user-plugin
    |   `-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
    |-- -zplg-find-completions-of-plugin
    |   `-- zplugin.zsh/-zplg-any-to-user-plugin
    |-- -zplg-format-env
    |-- -zplg-format-functions
    |-- -zplg-format-options
    |   |-- -zplg-restore-extendedglob
    |   `-- -zplg-save-set-extendedglob
    |-- -zplg-format-parameter
    `-- zplugin.zsh/-zplg-any-to-user-plugin

```

Called by:

```
zplugin.zsh/zplugin
```

-zplg-show-registered-plugins

Lists loaded plugins (subcommands list, loaded).

User-action entry point.

Has 21 line(s). Calls functions:

```

-zplg-show-registered-plugins
|-- -zplg-restore-extendedglob
|-- -zplg-save-set-extendedglob
`-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2

```

Called by:

```
zplugin.zsh/zplugin
```

-zplg-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:

```
-zplg-show-report
|-- -zplg-check-which-completions-are-enabled
|-- -zplg-check-which-completions-are-installed
|-- -zplg-diff-env-compute
|-- -zplg-diff-functions-compute
|-- -zplg-diff-options-compute
|-- -zplg-diff-parameter-compute
|-- -zplg-exists-message
|   |-- -zplg-any-to-uspl2
|   |   `-- zplugin.zsh/-zplg-any-to-user-plugin
|   `-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
|-- -zplg-find-completions-of-plugin
|   `-- zplugin.zsh/-zplg-any-to-user-plugin
|-- -zplg-format-env
|-- -zplg-format-functions
|-- -zplg-format-options
|   |-- -zplg-restore-extendedglob
|   `-- -zplg-save-set-extendedglob
|-- -zplg-format-parameter
`-- zplugin.zsh/-zplg-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
-zplg-show-all-reports
-zplg-show-debug-report
zplugin.zsh/zplugin
```

-zplg-show-times

Shows loading times of all loaded plugins.

User-action entry point.

Has 42 line(s). Calls functions:

```
-zplg-show-times  
'-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
```

Uses feature(s): *setopt*

Called by:

zplugin.zsh/zplugin

-zplg-show-zstatus

Shows Zplugin status, i.e. number of loaded plugins, of available completions, etc.

User-action entry point.

Has 41 line(s). Doesn't call other functions.

Uses feature(s): *setopt*

Called by:

zplugin.zsh/zplugin

-zplg-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 35 line(s). Calls functions:

```
-zplg-stress  
|-- zplugin-side.zsh/-zplg-exists-physically-message  
|-- zplugin-side.zsh/-zplg-first  
'-- zplugin.zsh/-zplg-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.).

-zplg-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:

```
-zplg-uncompile-plugin  
|-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2  
'-- zplugin.zsh/-zplg-any-to-user-plugin
```

Uses feature(s): *setopt*

Called by:

```
-zplg-compile-uncompile-all  
zplugin.zsh/zplugin
```

-zplg-uninstall-completions

Removes all completions of given plugin from Zshell (i.e. from FPATH).
The FPATH is typically `~/.zplugin/completions/`.

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

Has 41 line(s). Calls functions:

```
-zplg-uninstall-completions  
|-- -zplg-get-path  
|   |-- zplugin-side.zsh/-zplg-exists-physically  
|   |-- zplugin-side.zsh/-zplg-shands-exp  
|   |-- zplugin-side.zsh/-zplg-two-paths  
|   `-- zplugin.zsh/-zplg-any-to-user-plugin  
`-- zplugin-install.zsh/-zplg-forget-completion
```

Uses feature(s): *setopt*

Called by:

```
zplugin.zsh/zplugin
```

-zplg-unload

1. Call the Zsh Plugin's Standard *_plugin_unload function
2. Delete bindkeys (...)
3. Delete Zstyles
4. Restore options
5. Remove aliases
6. Restore Zle state
7. Unfunction functions (created by plugin)
8. Clean-up FPATH and PATH
9. Delete created variables
10. 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 347 line(s). Calls functions:

```
-zplg-unload
|-- -zplg-clear-report-for
|   '-- -zplg-any-to-uspl2
|       '-- zplugin.zsh/-zplg-any-to-user-plugin
|-- -zplg-diff-env-compute
|-- -zplg-diff-functions-compute
|-- -zplg-diff-options-compute
|-- -zplg-diff-parameter-compute
|-- -zplg-exists-message
|   |-- -zplg-any-to-uspl2
|   |   '-- zplugin.zsh/-zplg-any-to-user-plugin
|   '-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
|-- -zplg-restore-extendedglob
|-- -zplg-save-set-extendedglob
|-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
|-- zplugin.zsh/-zplg-any-to-user-plugin
|-- zplugin.zsh/-zplg-clear-debug-report
'-- zplugin.zsh/-zplg-unregister-plugin
```

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

Called by:

```
zplugin.zsh/-zplg-debug-unload
zplugin.zsh/-zplg-run-task
zplugin.zsh/zplugin
```

-zplg-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 212 line(s). Calls functions:

```
-zplg-update-or-status
|-- -zplg-at-eval
|-- -zplg-compute-ice
|   |-- zplugin-side.zsh/-zplg-exists-physically-message
|   |-- zplugin-side.zsh/-zplg-shands-exp
|   |-- zplugin-side.zsh/-zplg-two-paths
|   |-- zplugin.zsh/-zplg-any-to-user-plugin
|   `-- zplugin.zsh/-zplg-pack-ice
|-- -zplg-update-or-status-snippet
|   |-- -zplg-compute-ice
|   |   |-- zplugin-side.zsh/-zplg-exists-physically-message
|   |   |-- zplugin-side.zsh/-zplg-shands-exp
|   |   |-- zplugin-side.zsh/-zplg-two-paths
|   |   |-- zplugin.zsh/-zplg-any-to-user-plugin
|   |   `-- zplugin.zsh/-zplg-pack-ice
|   `-- zplugin.zsh/-zplg-load-snippet
|-- zplugin-install.zsh/-zplg-get-latest-gh-r-version
|-- zplugin-install.zsh/-zplg-setup-plugin-dir
|-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
|-- zplugin-side.zsh/-zplg-exists-physically-message
|-- zplugin-side.zsh/-zplg-store-ices
|-- zplugin-side.zsh/-zplg-two-paths
`-- zplugin.zsh/-zplg-any-to-user-plugin
```

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

Called by:


```
-zplg-update-or-status-all  
zplugin.zsh/zplugin
```

-zplg-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 63 line(s). Calls functions:

```

-zplg-update-or-status-all
|-- -zplg-update-or-status
|   |-- -zplg-at-eval
|   |-- -zplg-compute-ice
|       |-- zplugin-side.zsh/-zplg-exists-physically-message
|       |-- zplugin-side.zsh/-zplg-shands-exp
|       |-- zplugin-side.zsh/-zplg-two-paths
|       |-- zplugin.zsh/-zplg-any-to-user-plugin
|       `-- zplugin.zsh/-zplg-pack-ice
|   |-- -zplg-update-or-status-snippet
|       |-- -zplg-compute-ice
|           |-- zplugin-side.zsh/-zplg-exists-physically-message
|           |-- zplugin-side.zsh/-zplg-shands-exp
|           |-- zplugin-side.zsh/-zplg-two-paths
|           |-- zplugin.zsh/-zplg-any-to-user-plugin
|           `-- zplugin.zsh/-zplg-pack-ice
|       `-- zplugin.zsh/-zplg-load-snippet
|   |-- zplugin-install.zsh/-zplg-get-latest-gh-r-version
|   |-- zplugin-install.zsh/-zplg-setup-plugin-dir
|   |-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
|   |-- zplugin-side.zsh/-zplg-exists-physically-message
|   |-- zplugin-side.zsh/-zplg-store-ices
|   |-- zplugin-side.zsh/-zplg-two-paths
|   `-- zplugin.zsh/-zplg-any-to-user-plugin
|-- -zplg-update-or-status-snippet
|   |-- -zplg-compute-ice
|       |-- zplugin-side.zsh/-zplg-exists-physically-message
|       |-- zplugin-side.zsh/-zplg-shands-exp
|       |-- zplugin-side.zsh/-zplg-two-paths
|       |-- zplugin.zsh/-zplg-any-to-user-plugin
|       `-- zplugin.zsh/-zplg-pack-ice
|   `-- zplugin.zsh/-zplg-load-snippet
|-- zplugin-side.zsh/-zplg-any-colorify-as-uspl2
`-- zplugin.zsh/-zplg-any-to-user-plugin

```

Uses feature(s): *setopt*

Called by:

```
zplugin.zsh/zplugin
```

-zplg-update-or-status-snippet

Implements update or status operation for snippet given by URL.

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

Has 19 line(s). Calls functions:

```
-zplg-update-or-status-snippet
|-- -zplg-compute-ice
|   |-- zplugin-side.zsh/-zplg-exists-physically-message
|   |-- zplugin-side.zsh/-zplg-shands-exp
|   |-- zplugin-side.zsh/-zplg-two-paths
|   |-- zplugin.zsh/-zplg-any-to-user-plugin
|   `-- zplugin.zsh/-zplg-pack-ice
`-- zplugin.zsh/-zplg-load-snippet
```

Called by:

```
-zplg-update-or-status-all
-zplg-update-or-status
```

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 underscore (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*, *eval*, *read*, *setopt*, *unfunction*, *zle*, *zstyle*

Called by:

-zplg-compinit