

Systemd plugin

The systemd plugin provides many useful aliases for systemd.

To use it, add systemd to the plugins array of your zshrc file:

```
plugins=(... systemd)
```

Aliases

Alias	Command	Description
sc-list-units	systemctl list-units	List all units systemd has in memory
sc-is-active	systemctl is-active	Show whether a unit is active
sc-status	systemctl status	Show terse runtime status information about one or more units
sc-show	systemctl show	Show properties of units, jobs, or the manager itself
sc-help	systemctl help	Show man page of units
sc-list-unit-files	systemctl list-unit-files	List unit files installed on the system
sc-is-enabled	systemctl is-enabled	Checks whether any of the specified unit files are enabled
sc-list-jobs	systemctl list-jobs	List jobs that are in progress
sc-show-environment	systemctl show-environment	Dump the systemd manager environment block
sc-cat	systemctl cat	Show backing files of one or more units
sc-list-timers	systemctl list-timers	List timer units currently in memory
Aliases with sudo		
sc-start	sudo systemctl start	Start Unit(s)
sc-stop	sudo systemctl stop	Stop Unit(s)
sc-reload	sudo systemctl reload	Reload Unit(s)
sc-restart	sudo systemctl restart	Restart Unit(s)
sc-try-restart	sudo systemctl try-restart	Restart Unit(s)
sc-isolate	sudo systemctl isolate	Start a unit and its dependencies and stop all others
sc-kill	sudo systemctl kill	Kill unit(s)
sc-reset-failed	sudo systemctl reset-failed	Reset the "failed" state of the specified units,
sc-enable	sudo systemctl enable	Enable unit(s)

sc-disable	sudo systemctl disable	Disable unit(s)
sc-reenable	sudo systemctl reenable	Reenable unit(s)
sc-preset	sudo systemctl preset	Reset the enable/disable status one or more unit files
sc-mask	sudo systemctl mask	Mask unit(s)
sc-unmask	sudo systemctl unmask	Unmask unit(s)
sc-link	sudo systemctl link	Link a unit file into the unit file search path
sc-load	sudo systemctl load	Load unit(s)
sc-cancel	sudo systemctl cancel	Cancel job(s)
sc-set-environment	sudo systemctl set-environment	Set one or more systemd manager environment variables
sc-unset-environment	sudo systemctl unset-environment	Unset one or more systemd manager environment variables
sc-edit	sudo systemctl edit	Edit a drop-in snippet or a whole replacement file with <code>--full</code>
sc-enable-now	sudo systemctl enable --now	Enable and start unit(s)
sc-disable-now	sudo systemctl disable --now	Disable and stop unit(s)
sc-mask-now	sudo systemctl mask --now	Mask and stop unit(s)

User aliases

You can use the above aliases as `--user` by using the prefix `scu` instead of `sc`. For example: `scu-list-units` will be aliased to `systemctl --user list-units`.

Unit Status Prompt

You can add a token to your prompt in a similar way to the gitfast plugin. To add the token to your prompt, drop `$(systemd_prompt_info [unit]...)` into your prompt (more than one unit may be specified).

The plugin will add the following to your prompt for each `$unit`.

```
<prefix><unit>:<active|notactive><suffix>
```

You can control these parts with the following variables:

- `<prefix>` : Set `$ZSH_THEME_SYSTEMD_PROMPT_PREFIX`.
- `<suffix>` : Set `$ZSH_THEME_SYSTEMD_PROMPT_SUFFIX`.
- `<unit>` : name passed as parameter to the function. If you want it to be in ALL CAPS, you can set the variable `$ZSH_THEME_SYSTEMD_PROMPT_CAPS` to a non-empty string.
- `<active>` : shown if the systemd unit is active. Set `$ZSH_THEME_SYSTEMD_PROMPT_ACTIVE`.

- `<notactive>` : shown if the systemd unit is *not* active. Set `$ZSH_THEME_SYSTEMD_PROMPT_NOTACTIVE` .

For example, if your prompt contains `PROMPT='${systemd_prompt_info dhcpd httpd}'` and you set the following variables:

```
ZSH_THEME_SYSTEMD_PROMPT_PREFIX="[ "  
ZSH_THEME_SYSTEMD_PROMPT_SUFFIX="] "  
ZSH_THEME_SYSTEMD_PROMPT_ACTIVE="+"  
ZSH_THEME_SYSTEMD_PROMPT_NOTACTIVE="X"  
ZSH_THEME_SYSTEMD_PROMPT_CAPS=1
```

If `dhcpd` is running, and `httpd` is not, then your prompt will look like this:

```
[DHCPD: +][HTTPD: X]
```