

The Z Shell

<div> <div> Z shell @ Wikipedia </div> <div> Zsh home page </div> </div>	The Z Shell (zsh) is a POSIX compliant shell that is now the default shell of macOS and Kali Linux. <ul style="list-style-type: none"> This page is a quick reference to various topics related to the Z Shell. See PEL specific information here: 🔗 L - UNIX Shell and my USRHOME project for easy configuration of the Z Shell.
----------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Zsh Configuration

File	Users	Configuration file sourced order (See note 2)					Sourced for zsh -f	Similar to	Notes
		New Terminal session: Interactive Login	Sub-shell: Interactive non-login (includes Emacs shells)	Scripts : non-interactive, non-login	Scripts with special flag or piping output into SSH connection: Non-interactive login	Logout			Read section STARTUP/SHUTDOWN Files in the zsh(1) man page for information on configuration files. Use Emacs to display your local zsh man pages. See 🔗 Help/Info .
/etc/zshenv	All	1	1	1	1	Not sourced	Yes		
~/.zshenv	Current	2	2	2	2	Not sourced	No		<ul style="list-style-type: none"> Set ZOTDIR here to control location of other user zsh configuration files. Do NOT prepend append to PATH here! Otherwise they will be duplicated in sub-shells!
/etc/zprofile	All	3	Not sourced	Not sourced	3	Not sourced	No	~/.profile of: <ul style="list-style-type: none"> bash sh 	⚠ On macOS, /etc/zprofile does an eval of /usr/libexec/path_helper to set PATH.
\$ZDOTDIR/.zprofile	Current	4	Not sourced	Not sourced	4	Not sourced	No		<ul style="list-style-type: none"> Not sourced in sub-shell nor in script. However it is sourced on login. A good place to put logic that must be executed only once. <ul style="list-style-type: none"> If zsh is not your default shell, ~/.zprofile will not be sourced when you invoke the zsh command to launch a zsh sub-shell. Define environment variables that must be set once and not reset later when zsh is the default shell. Set EDITOR, prepend append user PATH here! Do NOT define alias and functions here, they will not be available in sub-shells!
/etc/zshrc	All	5	3	Not sourced	Not sourced	Not sourced	No		
\$ZDOTDIR/.zshrc	Current	6	4	Not sourced	Not sourced	Not sourced	No		<ul style="list-style-type: none"> Sourced in each sub-shells and script: put logic here that you want re-executed inside each sub-shell and script. <ul style="list-style-type: none"> If zsh is not the default shell, and something else like bash is, executing zsh from bash will only source ~/.zshenv and ~/.zshrc so if you want to redefine something that was defined in the Bash shell, it would be here. Define PROMPT, RPROMPT, anything that must check INSIDE_EMACS. The logic is sourced at login and when a sub-shell is created: the logic cannot prepend/append to envvar but can select values based on values of envvar set in sub-shells. Do NOT prepend append to PATH here! Otherwise they will be duplicated in sub-shells!
/etc/zlogin	All	7	Not sourced	Not sourced	5	Not sourced	No	~/.login of: <ul style="list-style-type: none"> ksh 	
\$ZDOTDIR/.zlogin	Current	8	Not sourced	Not sourced	6	Not sourced	No		
/etc/zlogout	All	Not sourced	Not sourced	Not sourced	Not sourced	1	No		
\$ZDOTDIR/.zlogout	Current	Not sourced	Not sourced	Not sourced	Not sourced	2	No		

Zsh - Notes

#	Topic	Note
1	ZDOTDIR	The location of the user specific configuration files can be modified by ZDOTDIR environment variable, set inside ~/.zshenv <ul style="list-style-type: none"> 👉 Instead of using the ZDOTDIR environment variable (which has to be set somewhere), you can also place symbolic links in the home directory that point to the location of the real files stored somewhere else. I prefer this method. It is simpler and works with all configuration files.
2	Configuration Files	<i>Note that none of the configuration files need a #!/bin/zsh shebang line.</i>
	Login	When opening a new login shell. For example when accessing the computer over ssh from another computer. 👉 On macOS, terminal emulators shells are login shells but they do authenticate (prompt) the user.
	Interactive shell	When executing zsh explicitly (or implicitly inside an editor like Emacs).
	Script	When executing a script that has a #!/bin/zsh shebang.
	Logout	Used as cleanup of login shell, executed before terminating the process of a login shell. If the shell is terminated by an external process these files might not run.
3	Envvar	Environment variables built-in shells that are quite useful.
	\$	Print the current process ID with: echo "\$\$" . The process ID of the shell parent process is \$PPID
	0	Name of the shell (really the name of the program that launched the shell). Print it with: echo "\$0"
	?	Exit code value of the last executed command. Print it with: echo "\$?" .
	SHLVL	An environment variable that represents the depth of a stack of shells executing each other. Supported by zsh, but also bash, ksh

Zsh - References

Topic	Reference
About Zsh	<ul style="list-style-type: none"> Z shell @ Wikipedia
Zsh Home	<ul style="list-style-type: none"> Zsh home page : with links to the following: <ul style="list-style-type: none"> Z shell page . : with links to information, documentation, support and source code pages.

Topic	Reference
Zsh intro, manual & tips	<ul style="list-style-type: none"> The zsh Manual: <ul style="list-style-type: none"> Zsh Manual (the manual in several formats) : the official zsh manual — current version. Introduction to Zsh (a November 30, 1995 document; gives an overview of Zsh early on). A User's Guide to the Z-Shell, by Peter Stephenson March 23, 2003. <ul style="list-style-type: none"> This also is a predecessor to the book From Bash to Z Shell published by APress in 2005 Moving to Zsh, from Armin Briegel, provides a good introduction to macOS users. Awesome-zsh lists several things zsh can do in an easy-to-read list format.
Zsh configuration info	<ul style="list-style-type: none"> How do Zsh Configuration Files Work? By Daniel Kehoe @ freeCodeCamp . Good into but 100% correct as it misses identifying some files. Zsh prompt expansion which lists the codes used to customize zsh prompt. Customizing the zsh Prompt , from Armin Briegel
Zsh configuration frameworks	<ul style="list-style-type: none"> Oh My Zsh - An open-source, community-driven framework for managing Zsh configuration. Pretzo Zimfw Zsh-snap
Other Zsh resources	<ul style="list-style-type: none"> Z-Shell.dev, a wiki-based collection of tools for zsh. Documentation is very scarce and would benefit from a overview description of the site's purpose. It seems to be devoted to Zi, but the purpose of Zi is not clear either. Their code is not well documented, an unfortunate trend in software these days. However some potentially useful pages might be useful: <ul style="list-style-type: none"> Zsh Plugin Standard — seems to list a set of conventions - at this point I'm not sure if these are followed globally or it's just for Zi. Zsh Native Scripting Handbook — a list of Zsh script snippets.
Zsh man pages	<p>👉 Use Emacs to display zsh man pages! Viewing man pages with Emacs gives you great navigation and control.</p> <ul style="list-style-type: none"> See 📄 Help/Info toward end of page 5 for the commands to open a man page. Use M-⌘8 with PEL.
Zsh Linters	High-quality linter for Zsh do not seem to exist. Shellcheck mostly supports bash and sh, not zsh specifics. The other two Zsh linters I found seem to be in early or abandoned state: zlint and zsh-lint .
Other Topics	<ul style="list-style-type: none"> Reload Zsh configuration - Bozhidar Batsov article