ShellCheck Shell script linter

CONCEPTS

Every finding has an SC code. Search it on the ShellCheck wiki (github.com/koalaman/shellcheck/wiki/SC2086) for the reason and the fix.

Severity order is error, warning, info, style. Exit code 0 means clean, 1 means findings, 2 means a file could not be read.

RUNNING

`shellcheck <file>...`
Check one or more scripts

`shellcheck -`
Read the script from stdin

`shellcheck -x <file>`
Also follow files it sources

`find . -name '*.sh' -exec shellcheck {} +`
Check every script in a tree

`shellcheck --version`
Show the installed version

DIALECT AND CHECKS

`-s <shell>`
Force sh, bash, dash, ksh or busybox

`-S <severity>`
Minimum level: error, warning, info, style

`-e <codes>`
Exclude codes, comma separated

OUTPUT AND EXIT CODES

`-f tty`
Readable text with code excerpts (default)

`-f gcc`
file:line:col lines for editors

`-f json1`
Machine-readable JSON with a comments array

OUTPUT AND EXIT CODES (cont.)

`-f diff`
Unified diff of automatic fixes

`SHELLCHECK_OPTS="--shell=bash -e SC2016"`
Default flags for every run

DIRECTIVES

`# shellcheck disable=SC2086`
Ignore a code for the next command

`# shellcheck source=<path>`
Tell it which file a source loads

`# shellcheck shell=bash`
Set the dialect for this file

A directive on its own line before the first command applies to the whole file; anywhere else it covers only the next command or block.

CONFIG FILES

`.shellcheckrc`
Project config, found in script dir or parents

`disable=SC2034`
rc: ignore a code everywhere

`external-sources=true`
rc: always follow sourced files

COMMON WARNINGS

`SC2086`
Quote $var to stop splitting and globbing

`SC2046`
Quote $(...) to stop word splitting

`SC2034`
Variable assigned but never used

`SC2164`
Use cd ... || exit in case cd fails

`SC1090 / SC1091`
Cannot follow source; use -x or a source= directive