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.

Script file or stdin
↓ parse
Shell dialect -s, directive, shebang
↓ check
Findings SC code + severity
↓ filter
Report -f format, exit code

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

apt install shellcheck
Install on Debian/Ubuntu

brew install shellcheck
Install on macOS

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

-i *<codes>*
Report only these codes

-o *<names>*
Enable optional checks

DIALECT AND CHECKS (cont.)

-P *<dirs>*
Search path for sourced files

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

-f diff
Unified diff of automatic fixes

echo $?
Exit code: 0 clean, 1 findings

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

DIRECTIVES

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

 # shellcheck disable=SC2034,SC2154
Ignore several codes

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

 # shellcheck source=/dev/null
Do not follow this source

 # shellcheck source-path=SCRIPTDIR
Look up sources next to the script

 # 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

CONFIG FILES (cont.)

shell=bash
rc: default dialect

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

SC2068
Use "$@" instead of unquoted $@

SC2034
Variable assigned but never used

SC2154
Variable used but never assigned

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

SC2155
Declare and assign separately

SC2162
read without -r mangles backslashes

SC2006
Use $(...) instead of backticks

SC2016
Single quotes do not expand variables

SC2012
Do not parse ls; use a glob or find

SC2002
Useless cat; redirect with < file

SC2015
A && B || C is not if-then-else

SC2181
Test the command directly, not $?

SC2148
Add a shebang or a shell directive

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