

SS64

Linux >

How-to >

Search

shopt

Shell Options

Syntax

```
shopt [-pqsu] [-o] [optname ...]
```

Options

- s Enable (set) each *optname*
- u Disable (unset) each *optname*.
- p Display a list of all settable options, with an indication of whether or not each is set. The output is displayed in a form that can be reused as input. (-p is the default action)
- q Suppresses normal output; the return status indicates whether the *optname* is set or unset. If multiple *optname* arguments are given with '-q', the return status is zero if all *optnames* are enabled; non-zero otherwise.
- o Restricts the values of *optname* to be those defined for the '-o' option to the set builtin.

If either '-s' or '-u' is used with no *optname* arguments, the display is limited to those options which are set or unset, respectively.

Unless otherwise noted, the shopt options are disabled (off) by default.

The return status is zero unless an *optname* was **not** enabled or was not a valid shell option.

optnames:

cdable_vars

If this is set, an argument to the cd builtin command that is not a directory is assumed to be the name of a variable whose value is the directory to change to.

cdspell

If set, minor errors in the spelling of a directory component in a cd command will be corrected. The errors checked for are transposed characters, a missing character, and a character too many. If a correction is found, the corrected path is printed, and the command proceeds. This option is only used by interactive shells.

checkhash

If this is set, Bash checks that a command found in the hash table exists before trying to execute it. If a hashed command no longer exists, a normal path search is performed.

checkwinsize

If set, Bash checks the window size after each command and, if necessary, updates the values of LINES and COLUMNS.

cmdhist

If set, Bash attempts to save all lines of a multiple-line command in the same history entry. This allows easy re-editing of multi-line commands.

dotglob

If set, Bash includes filenames beginning with a '.' in the results of filename expansion.

execfail

If this is set, a non-interactive shell will not exit if it cannot execute the file specified as an argument to the exec builtin command. An interactive shell does not exit if exec fails.

expand_aliases

If set, aliases are expanded.
This option is enabled by default for interactive shells.

extglob

If set, the extended pattern matching features described above are enabled.

histappend

If set, the history list is appended to the history file when the shell exits, rather than overwriting the history file.

```
shopt -s histappend
```

To append every line to history individually set:

PROMPT_COMMAND='history -a'

With these two settings, a new shell will get the history lines from all previous shells instead of the default 'last window closed'>history
(the history file is named by the value of the HISTFILE variable)

histreedit

If set, and Readline is being used, a user is given the opportunity to re-edit a failed history substitution.

histverify

If set, and Readline is being used, the results of history substitution are not immediately passed to the shell parser. Instead, the resulting line is loaded into the Readline editing buffer, allowing further modification.

hostcomplete

If set, and Readline is being used, Bash will attempt to perform hostname completion when a word containing a '@' is being completed.
This option is enabled by default.

huponexit

If set, Bash will send **SIGHUP** to all jobs when an interactive login shell exits.

interactive_comments

Allow a word beginning with '#' to cause that word and all remaining characters on that line to be ignored in an interactive shell.
This option is enabled by default.

lithist

If enabled, and the cmdhist option is enabled, multi-line commands are saved to the history with embedded newlines rather than using semicolon separators where possible.

mailwarn

If set, and a file that Bash is checking for mail has been accessed since the last time it was checked, the message "The mail in *mailfile* has been read" is displayed.

no_empty_cmd_completion

If set, and Readline is being used, Bash will not attempt to search the PATH for possible completions when completion is attempted on an empty line.

nocaseglob

If set, Bash matches filenames in a case-insensitive fashion when performing filename expansion.

nullglob

If set, Bash allows filename patterns which match no files to expand to a null string, rather than themselves.

progcomp

If set, the programmable completion facilities are enabled.
This option is enabled by default.

promptvars

If set, prompt strings undergo variable and parameter expansion after being expanded. This option is enabled by default.

restricted_shell

The shell sets this option if it is started in restricted mode. The value can not be changed. This is not reset when the startup files are executed, allowing the startup files to discover whether or not a shell is restricted.

shift_verbose

If this is set, the shift builtin prints an error message when the shift count exceeds the number of positional parameters.

sourcepath

If set, the source builtin uses the value of PATH to find the directory containing the file supplied as an argument. This option is enabled by default.

xpg_echo

If set, the echo builtin expands backslash-escape sequences by default.

'shopt' is a BASH builtin command.

Examples

Turn on spell checker for directory names in the **cd** builtin:

```
$ shopt -s cdspell
```

"The dogma of the ghost in the machine" ~ Gilbert Ryle

Related linux commands

[env](#) - Display, set, or remove environment variables.

[function](#) - Define Function Macros.

[local](#) - Create variables.

[logname](#) - Print current login name.

[set](#) - Manipulate shell variables and functions.

[uname](#) - Print system information.

[users](#) - Print login names of users currently logged in.

[who](#) - Print who is currently logged in.

Equivalent Windows commands: [SETLOCAL](#) - Set options to control the visibility of variables.

