otherwise 0.

shopt [**-pqsu**] [**-o**] [*optname* ...]
Toggle the values of settings controlling optional shell behavior. The settings can be either those listed below, or, if the **-o** option is used, those available with the **-o** option to the **set** builtin command. With no options, or with the **-p** option, a list of all settable options is displayed, with an indication of whether or not each is set; if *optnames* are supplied, the output is restricted to those options. The **-p** option causes output to be displayed in a form that may be reused as input. Other options have the following meanings:

-s Enable (set) each *optname*.

-u Disable (unset) each *optname*.

-q Suppresses normal output (quiet mode); 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, **shopt** shows only those options which are set or unset, respectively. Unless otherwise noted, the **shopt** options are disabled (unset) by default.

The return status when listing options is zero if all *optnames* are enabled, non-zero otherwise. When setting or unsetting options, the return status is zero unless an *optname* is not a valid shell option.

The list of **shopt** options is:

assoc_expand_once
If set, the shell suppresses multiple evaluation of associative array subscripts during arithmetic expression evaluation, while executing builtins that can perform variable assignments, and while executing builtins that perform array dereferencing.

autocd If set, a command name that is the name of a directory is executed as if it were the argument to the **cd** command. This option is only used by interactive shells.

cdable_vars
If 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 one character too many. If a correction is found, the corrected filename is printed, and the command proceeds. This option is only used by interactive shells.

checkhash
If 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.

checkjobs
If set, **bash** lists the status of any stopped and running jobs before exiting an interactive shell. If any jobs are running, this causes the exit to be deferred until a second exit is attempted without an intervening command (see **JOB CONTROL** above). The shell always postpones exiting if any jobs are stopped.

checkwinsize
If set, **bash** checks the window size after each external (non-builtin) command and, if necessary, updates the values of **LINES** and **COLUMNS**. This option is enabled by default.

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. This option is enabled by default, but only has an effect if command history is enabled, as described above under **HISTORY**.

compat31
If set, **bash** changes its behavior to that of version 3.1 with respect to quoted arguments to the **[[** conditional command's **=~** operator and locale-specific string comparison when using the **[[** conditional command's **<** and **>** operators. Bash versions prior to bash-4.1 use ASCII collation and strcmp(3); bash-4.1 and later use the current locale's collation sequence and strcoll(3).

compat32
If set, **bash** changes its behavior to that of version 3.2 with respect to locale-specific string comparison when using the **[[** conditional command's **<** and **>** operators (see previous item) and the effect of interrupting a command list. Bash versions 3.2 and earlier continue with the next command in the list after one terminates due to an interrupt.

compat40
If set, **bash** changes its behavior to that of version 4.0 with respect to locale-specific string comparison when using the **[[** conditional command's **<** and **>** operators (see description of **compat31**) and the effect of interrupting a command list. Bash versions 4.0 and later interrupt the list as if the shell received the interrupt; previous versions continue with the next command in the list.

compat41
If set, **bash**, when in *posix mode*, treats a single quote in a double-quoted parameter expansion as a special character. The single quotes must match (an even number) and the characters between the single quotes are considered quoted. This is the behavior of posix mode through version 4.1. The default bash behavior remains as in previous versions.

compat42

If set, **bash** does not process the replacement string in the pattern substitution word expansion using quote removal.

compat43
If set, **bash** does not print a warning message if an attempt is made to use a quoted compound array assignment as an argument to **declare**, makes word expansion errors non-fatal errors that cause the current command to fail (the default behavior is to make them fatal errors that cause the shell to exit), and does not reset the loop state when a shell function is executed (this allows **break** or **continue** in a shell function to affect loops in the caller's context).

compat44
If set, **bash** saves the positional parameters to BASH_ARGV and BASH_ARGC before they are used, regardless of whether or not extended debugging mode is enabled.

complete_fullquote
If set, **bash** quotes all shell metacharacters in filenames and directory names when performing completion. If not set, **bash** removes metacharacters such as the dollar sign from the set of characters that will be quoted in completed filenames when these metacharacters appear in shell variable references in words to be completed. This means that dollar signs in variable names that expand to directories will not be quoted; however, any dollar signs appearing in filenames will not be quoted, either. This is active only when bash is using backslashes to quote completed filenames. This variable is set by default, which is the default bash behavior in versions through 4.2.

direxpand
If set, **bash** replaces directory names with the results of word expansion when performing filename completion. This changes the contents of the readline editing buffer. If not set, **bash** attempts to preserve what the user typed.

dirspell
If set, **bash** attempts spelling correction on directory names during word completion if the directory name initially supplied does not exist.

dotglob If set, **bash** includes filenames beginning with a `.' in the results of pathname expansion. The filenames ``.'' and ``..'' must always be matched explicitly, even if **dotglob** is set.

execfail
If 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 as described above under **ALIASES**. This option is enabled by default for interactive shells.

extdebug

If set at shell invocation, arrange to execute the debugger profile before the shell starts, identical to the **--debugger** option. If set after invocation, behavior intended for use by debuggers is enabled:

1. The **-F** option to the **declare** builtin displays the source file name and line number corresponding to each function name supplied as an argument.
2. If the command run by the **DEBUG** trap returns a non-zero value, the next command is skipped and not executed.
3. If the command run by the **DEBUG** trap returns a value of 2, and the shell is executing in a subroutine (a shell function or a shell script executed by the **.** or **source** builtins), the shell simulates a call to **return**.
4. **BASH_ARGC** and **BASH_ARGV** are updated as described in their descriptions above.
5. Function tracing is enabled: command substitution, shell functions, and subshells invoked with **(** *command* **)** inherit the **DEBUG** and **RETURN** traps.
6. Error tracing is enabled: command substitution, shell functions, and subshells invoked with **(** *command* **)** inherit the **ERR** trap.

extglob If set, the extended pattern matching features described above under **Pathname Expansion** are enabled.

extquote
If set, **$**'*string*' and **$**"*string*" quoting is performed within **${***parameter***}** expansions enclosed in double quotes. This option is enabled by default.

failglob
If set, patterns which fail to match filenames during pathname expansion result in an expansion error.

force_fignore
If set, the suffixes specified by the **FIGNORE** shell variable cause words to be ignored when performing word completion even if the ignored words are the only possible completions. See **SHELL VARIABLES** above for a description of **FIGNORE**. This option is enabled by default.

globasciiranges
If set, range expressions used in pattern matching bracket expressions (see **Pattern Matching** above) behave as if in the traditional C locale when performing comparisons. That is, the current locale's collating sequence is not taken into account, so **b** will not collate between **A** and **B**, and upper-case and lower-case ASCII characters will collate together.

globstar
If set, the pattern ****** used in a pathname expansion context will match all files and zero or more directories and subdirectories. If the pattern is followed by a **/**, only directories and subdirectories match.

gnu_errfmt
If set, shell error messages are written in the

standard GNU error message format.

histappend
If set, the history list is appended to the file named by the value of the **HISTFILE** variable when the shell exits, rather than overwriting the file.

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 (see **Completing** under **READLINE** above). This is enabled by default.

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

inherit_errexit
If set, command substitution inherits the value of the **errexit** option, instead of unsetting it in the subshell environment. This option is enabled when *posix mode* is enabled.

interactive_comments
If set, allow a word beginning with **#** to cause that word and all remaining characters on that line to be ignored in an interactive shell (see **COMMENTS** above). This option is enabled by default.

lastpipe
If set, and job control is not active, the shell runs the last command of a pipeline not executed in the background in the current shell environment.

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

localvar_inherit
If set, local variables inherit the value and attributes of a variable of the same name that exists at a previous scope before any new value is assigned. The nameref attribute is not inherited.

localvar_unset
If set, calling **unset** on local variables in previous function scopes marks them so subsequent lookups find them unset until that function returns. This is identical to the behavior of unsetting local variables at the current function scope.

login_shell
The shell sets this option if it is started as a login shell (see **INVOCATION** above). The value may not be changed.

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 pathname expansion (see **Pathname Expansion** above).

nocasematch
If set, **bash** matches patterns in a case-insensitive fashion when performing matching while executing **case** or **[[** conditional commands, when performing pattern substitution word expansions, or when filtering possible completions as part of programmable completion.

nullglob
If set, **bash** allows patterns which match no files (see **Pathname Expansion** above) to expand to a null string, rather than themselves.

progcomp
If set, the programmable completion facilities (see **Programmable Completion** above) are enabled. This option is enabled by default.

progcomp_alias
If set, and programmable completion is enabled, **bash** treats a command name that doesn't have any completions as a possible alias and attempts alias expansion. If it has an alias, **bash** attempts programmable completion using the command word resulting from the expanded alias.

promptvars
If set, prompt strings undergo parameter expansion, command substitution, arithmetic expansion, and quote removal after being expanded as described in **PROMPTING** above. This option is enabled by default.

restricted_shell
The shell sets this option if it is started in restricted mode (see **RESTRICTED SHELL** below). The value may 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 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.