**NAME**

shell_builtins – shell command interpreter built-in functions

**DESCRIPTION**

The shell command interpreters (**sh**(1), **csh**(1), and **ksh**(1)), have special built-in functions which are interpreted by the shell as commands. Many of these built-in commands are implemented by more than one of the shells, and some are unique to a particular shell. These are:

| **command** | **built into** |
|---|---|
| ------- | ----- |
| **alias** | csh, ksh |
| **bg** | csh, ksh, sh |
| **break** | csh, ksh, sh |
| **case** | csh, ksh, sh |
| **cd** | csh, ksh, sh |
| **chdir** | csh, sh |
| **continue** | csh, ksh, sh |
| **dirs** | csh |
| **echo** | csh, ksh, sh |
| **eval** | csh, ksh, sh |
| **exec** | csh, ksh, sh |
| **exit** | csh, ksh, sh |
| **export** | ksh, sh |
| **fc** | ksh |
| **fg** | csh, ksh, sh |
| **for** | ksh, sh |
| **foreach** | csh |
| **function** | ksh |
| **getopts** | ksh, sh |
| **glob** | csh |
| **goto** | csh |
| **hash** | ksh, sh |
| **hashstat** | csh |
| **history** | csh |
| **if** | csh, ksh, sh |
| **jobs** | csh, ksh, sh |
| **kill** | csh, ksh, sh |
| **let** | ksh |
| **limit** | csh |
| **login** | csh, ksh, sh |
| **logout** | csh, ksh, sh |
| **nice** | csh |
| **newgrp** | ksh, sh |
| **notify** | csh |
| **onintr** | csh |

| | |
|---|---|
| **popd** | csh |
| **print** | ksh |
| **pushd** | csh |
| **pwd** | ksh, sh |
| **read** | ksh, sh |
| **readonly** | ksh, sh |
| **rehash** | csh |
| **repeat** | csh |
| **return** | ksh, sh |
| **select** | ksh |
| **set** | csh, ksh, sh |
| **setenv** | csh |
| **shift** | csh, ksh, sh |
| **source** | csh |
| **stop** | csh, ksh, sh |
| **suspend** | csh, ksh, sh |
| **switch** | csh |
| **test** | ksh, sh |
| **time** | csh |
| **times** | ksh, sh |
| **trap** | ksh, sh |
| **type** | ksh, sh |
| **typeset** | ksh |
| **ulimit** | ksh, sh |
| **umask** | csh, ksh, sh |
| **unalias** | csh, ksh |
| **unhash** | csh |
| **unlimit** | csh |
| **unset** | csh, ksh, sh |
| **unsetenv** | csh |
| **until** | ksh, sh |
| **wait** | csh, ksh, sh |
| **whence** | ksh |
| **while** | csh, ksh, sh |

**Bourne Shell, sh, Special Commands**

Input/output redirection is now permitted for these commands. File descriptor 1 is the default output location. When Job Control is enabled, additional *Special Commands* are added to the shell's environment.

Additional to these built-in reserved command words, **sh** also uses:

**:** No effect; the command does nothing. A zero exit code is returned.

**.** *filename*
Read and execute commands from *filename* and return. The search path specified by **PATH** is used to find the directory containing *filename*.

**C shell, csh**

Built-in commands are executed within the C shell. If a built-in command occurs as any component of a pipeline except the last, it is executed in a subshell. Additional to these built-in reserved command words, **csh** also uses:

**:** Null command. This command is interpreted, but performs no action.

**Korn Shell, ksh, Special Commands**

Input/Output redirection is permitted. Unless otherwise indicated, the output is written on file descriptor 1 and the exit status, when there is no syntax error, is zero.

Commands that are preceded by one or two † (daggers) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by †† that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the **=** sign and word splitting and file name generation are not performed.

Additional to these built-in reserved command words, **ksh** also uses:

† **:** [ *arg* . . . ]
The command only expands parameters.

† **.** *file* **[** *arg* **. . . ]**
Read the complete *file* then execute the commands. The commands are executed in the current shell environment. The search path specified by **PATH** is used to find the directory containing *file*. If any arguments *arg* are given, they become the positional parameters. Otherwise the positional parameters are unchanged. The exit status is the exit status of the last command executed. the loop termination test.

**SEE ALSO**

**alias**(1), **break**(1), **case**(1), **cd**(1), **chmod**(1), **csh**(1), **echo**(1), **exec**(1), **exit**(1), **for**(1), **find**(1), **function**(1), **getoptcvt**(1) **getopts**(1), **glob**(1), **hash**(1), **history**(1), **if**(1), **intro**(1), **jobs**(1), **kill**(1), **ksh**(1), **let**(1), **limit**(1), **login**(1), **logout**(1), **newgrp**(1), **nice**(1), **nohup**(1), **print**(1), **pwd**(1), **read**(1), **readonly**(1), **repeat**(1), **set**(1), **sh**(1), **shift**(1), **suspend**(1), **test**(1B), **time**(1), **times**(1), **trap**(1), **typeset**(1), **umask**(1), **wait**(1), **while**(1), **chdir**(2), **chmod**(2), **creat**(2), **umask**(2), **getopt**(3C), **profile**(4), **environ**(5)