NAME

getopt – parse command options

SYNOPSIS

set —— `getopt** *optstring* **$*`

AVAILABILITY

SUNWcsu

DESCRIPTION

The **getopts** command supersedes **getopt**. For more information, see **NOTES** below.

getopt is used to break up options in command lines for easy parsing by shell procedures and to check for legal options. *optstring* is a string of recognized option letters; see **getopt**(3C). If a letter is followed by a colon, the option is expected to have an argument which may or may not be separated from it by white space. The special option —— is used to delimit the end of the options. If it is used explicitly, **getopt** recognizes it; otherwise, **getopt** generates it; in either case, **getopt** places it at the end of the options. The positional parameters (**$1 $2** . . .) of the shell are reset so that each option is preceded by a – and is in its own positional parameter; each option argument is also parsed into its own positional parameter.

EXAMPLES

The following code fragment shows how one might process the arguments for a command that can take the options **a** or **b**, as well as the option **o**, which requires an argument:

```
set -- `getopt abo: $*`
if [ $? != 0 ]
then
        echo $USAGE
        exit 2
fi
for i in $*
do
        case $i in
        -a | -b)        FLAG=$i; shift;;
        -o)             OARG=$2; shift 2;;
        --)             shift; break;;
        esac
done
```

This code accepts any of the following as equivalent:

```
cmd -aoarg filename1 filename2
cmd -a -o arg filename1 filename2
cmd -oarg -a filename1 filename2
cmd -a -oarg -- filename1 filename2
```

SEE ALSO

intro(1), **shell_builtins**(1), **sh**(1), **getopt**(3C)

DIAGNOSTICS

getopt prints an error message on the standard error when it encounters an option letter not included in *optstring*.

NOTES

getopt will not be supported in the next major release. For this release a conversion tool has been provided, **getoptcvt**. For more information about **getopts** and **getoptcvt**, see **getopts**(1).

Reset **optind** to 1 when rescanning the options.

getopt does not support the part of Rule 8 of the command syntax standard (see **intro**(1)) that permits groups of option-arguments following an option to be separated by white space and quoted. For example,

cmd −a −b −o "xxx z yy" filename

is not handled correctly. To correct this deficiency, use the **getopts** command in place of **getopt**.

If an option that takes an option-argument is followed by a value that is the same as one of the options listed in *optstring* (referring to the earlier **EXAMPLES** section, but using the following command line: **cmd -o -a filename**, **getopt** always treats −**a** as an option-argument to −**o**; it never recognizes −**a** as an option. For this case, the **for** loop in the example shifts past the *filename* argument.