

NAME

errc, verrc, warnc, vwarnc — formatted error messages

LIBRARY

Utility functions from BSD systems (libbsd, -lbsd)

SYNOPSIS

```
#include <err.h>
```

(See *libbsd(7)* for include usage.)

```
void
```

```
errc(int status, int code, const char *fmt, ...);
```

```
void
```

```
verrc(int status, int code, const char *fmt, va_list args);
```

```
void
```

```
warnc(int code, const char *fmt, ...);
```

```
void
```

```
vwarnc(int code, const char *fmt, va_list args);
```

DESCRIPTION

The **err()** and **warn()** family of functions display a formatted error message on the standard error output. In all cases, the last component of the program name, followed by a colon (':') character and a space, are output. The text that follows depends on the function being called. The *fmt* specification (and associated arguments) may be any format allowed by *printf(3)* or NULL. If the *fmt* argument is not NULL, the formatted error message is output.

The functions all output an error message string affiliated with an error value (see *strerror(3)*), preceded by a colon character and a space if *fmt* is not NULL. That is, the output is as follows:

```
    progname: fmt: error message string
```

if *fmt* is not NULL, or:

```
    progname: error message string
```

if it is.

The argument *code* is used as the error value instead of the current value of the global variable *errno*.

In all cases, the output is followed by a newline character.

The **errc()**, and **verrc()** functions do not return, but exit with the value of the argument *status*.

EXAMPLES

Display the current *errno* information string and exit:

```
if ((p = malloc(size)) == NULL)
    err(1, NULL);
if ((fd = open(file_name, O_RDONLY, 0)) == -1)
    err(1, "%s", file_name);
```

Display an error message and exit:

```
if (tm.tm_hour < START_TIME)
    errx(1, "too early, wait until %s", start_time_string);
```

Warn of an error:

```
if ((fd = open(raw_device, O_RDONLY, 0)) == -1)
    warnx("%s: %s: trying the block device",
        raw_device, strerror(errno));
if ((fd = open(block_device, O_RDONLY, 0)) == -1)
    err(1, "%s", block_device);
```

SEE ALSO

err(3) *exit(3)*, *perror(3)*, *printf(3)*, *strerror(3)*

HISTORY

The functions **errc()**, **verrc()**, **warnc()**, and **vwarnc()** first appeared in FreeBSD 3.0, NetBSD 7.0 and OpenBSD 5.6.

CAVEATS

It is important never to pass a string with user-supplied data as a format without using ‘%S’. An attacker can put format specifiers in the string to mangle the stack, leading to a possible security hole. This holds true even if the string has been built “by hand” using a function like **snprintf()**, as the resulting string may still contain user-supplied conversion specifiers for later interpolation by the **err()** and **warn()** family of functions.

Always be sure to use the proper secure idiom:

```
errc(1, 0, "%s", string);
```