NAME

err, **verr**, **errc**, **verrc**, **errx**, **verrx**, **warn**, **vwarn**, **warnc**, **vwarnc**, **warnx**, **vwarnx**, **err_set_exit**, **err_set_file** - formatted error messages

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <err.h>

void
err(*int eval, const char *fmt, ...*);

void
err_set_exit(*void (*exitf)(int)*);

void
err_set_file(*void *vfp*);

void
errc(*int eval, int code, const char *fmt, ...*);

void
errx(*int eval, const char *fmt, ...*);

void
warn(*const char *fmt, ...*);

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

void
warnx(*const char *fmt, ...*);

#include <stdarg.h>

void
verr(*int eval, const char *fmt, va_list args*);

void

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

void
verrx(*int eval, const char *fmt, va_list args*);

void
vwarn(*const char *fmt, va_list args*);

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

void
vwarnx(*const char *fmt, va_list args*);

DESCRIPTION

The **err**() and **warn**() family of functions display a formatted error message on the standard error output, or on another file specified using the **err_set_file**() function. In all cases, the last component of the program name, a colon character, and a space are output. If the *fmt* argument is not NULL, the printf(3)-like formatted error message is output. The output is terminated by a newline character.

The **err**(), **errc**(), **verr**(), **verrc**(), **warn**(), **warnc**(), **vwarn**(), and **vwarnc**() functions append an error message obtained from strerror(3) based on a supplied error code value or the global variable *errno*, preceded by another colon and space unless the *fmt* argument is NULL.

In the case of the **errc**(), **verrc**(), **warnc**(), and **vwarnc**() functions, the *code* argument is used to look up the error message.

The **err**(), **verr**(), **warn**(), and **vwarn**() functions use the global variable *errno* to look up the error message.

The **errx**() and **warnx**() functions do not append an error message.

The **err**(), **verr**(), **errc**(), **verrc**(), **errx**(), and **verrx**() functions do not return, but exit with the value of the argument *eval*. It is recommended that the standard values defined in sysexits(3) be used for the value of *eval*. The **err_set_exit**() function can be used to specify a function which is called before exit(3) to perform any necessary cleanup; passing a null function pointer for *exitf* resets the hook to do nothing. The **err_set_file**() function sets the output stream used by the other functions. Its *vfp* argument must be either a pointer to an open stream (possibly already converted to void *) or a null pointer (in which case the output stream is set to standard error).

EXAMPLES

Display the current errno information string and exit:

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

Display an error message and exit:

```
if (tm.tm_hour < START_TIME)
        errx(EX_DATAERR, "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(EX_OSFILE, "%s", block_device);
```

Warn of an error without using the global variable *errno*:

```
error = my_function();          /* returns a value from <errno.h> */
if (error != 0)
        warnc(error, "my_function");
```

SEE ALSO

exit(3), fmtmsg(3), printf(3), strerror(3), sysexits(3)

STANDARDS

The **err**() and **warn**() families of functions are BSD extensions. As such they should not be used in truly portable code. Use **strerror**() or similar functions instead.

HISTORY

The **err**() and **warn**() functions first appeared in 4.4BSD. The **err_set_exit**() and **err_set_file**() functions first appeared in FreeBSD 2.1. The **errc**() and **warnc**() functions first appeared in FreeBSD 3.0.