

NAME

addstr, **waddstr**, **mvaddstr**, **mvwaddstr**, **addnstr**, **waddnstr**, **mvaddnstr**, **mvwaddnstr** – add a string to a *curses* window and advance the cursor

SYNOPSIS

```
#include <curses.h>
```

```
int addstr(const char * str);
int waddstr(WINDOW * win, const char * str);
int mvaddstr(int y, int x, const char * str);
int mvwaddstr(WINDOW * win, int y, int x, const char * str);

int addnstr(const char * str, int n);
int waddnstr(WINDOW * win, const char * str, int n);
int mvaddnstr(int y, int x, const char * str, int n);
int mvwaddnstr(WINDOW * win, int y, int x, const char * str, int n);
```

DESCRIPTION

waddstr writes the characters of the (null-terminated) string *str* to the window *win*, as if by calling **waddch(3X)** for each *char* in *str*. **waddnstr** is similar, but writes at most *n* characters. If *n* is -1 , **waddnstr** writes the entire string. **ncurses(3X)** describes the variants of these functions.

RETURN VALUE

These functions return **OK** on success and **ERR** on failure.

In *ncurses*, these functions fail if

- the *curses* screen has not been initialized,
- *str* is a null pointer,
- (for functions taking a *WINDOW* pointer argument) *win* is a null pointer, or
- an internal **waddch(3X)** call returns **ERR**.

Functions prefixed with “mv” first perform cursor movement and fail if the position (*y*, *x*) is outside the window boundaries.

NOTES

All of these functions except **waddnstr** may be implemented as macros.

PORTABILITY

X/Open Curses Issue 4 describes these functions. It specifies no error conditions for them.

SVr4 describes a successful return value only as “an integer value other than *ERR*”.

HISTORY

4BSD (1980) introduced *waddstr* along with its variants, the latter defined as macros.

SVr3.1 (1987) added *waddnstr* (and its variants) redefining *waddstr* as a macro wrapping it.

SEE ALSO

curs_addwstr(3X) describes comparable functions of the *ncurses* library in its wide-character configuration (*ncursesw*).

curses(3X), **curs_addch(3X)**, **curs_addchstr(3X)**