NAME

ins_wstr, **wins_wstr**, **mvins_wstr**, **mvwins_wstr**, **ins_nwstr**, **wins_nwstr**, **mvins_nwstr**, **mvwins_nwstr** − insert a wide-character string in a*cur ses* window

SYNOPSIS

#include <curses.h>

int ins_wstr(const wchar_t * *wstr***);**
int wins_wstr(WINDOW * *win***, const wchar_t *** *wstr***);**
int mvins_wstr(int *y***, int** *x***, const wchar_t *** *wstr***);**
int mvwins_wstr(WINDOW * *win***, int** *y***, int** *x***,**
const wchar_t * *wstr***);**

int ins_nwstr(const wchar_t * *wstr***, int** *n***);**
int wins_nwstr(WINDOW * *win***, const wchar_t *** *wstr***, int** *n***);**
int mvins_nwstr(int *y***, int** *x***, const wchar_t *** *wstr***, int** *n***);**
int mvwins_nwstr(WINDOW * *win***, int** *y***, int** *x***,**
const wchar_t * *wstr***, int** *n***);**

DESCRIPTION

wins_wstr inserts a wide-character string *wstr* before the character at the cursor in window *win* as if by calling **wins_wch**(3X) for each *wchar_t* in *wstr*. No line wrapping is performed. Characters to the right of the cursor are shifted right; those at the right edge of the window may be lost. **wins_wstr** stops inserting if it would have to wrap to the next line to write the next *wchar_t* in *wstr*. The cursor position does not change (after moving to (*y*, *x*), if specified). **wins_nwstr** does the same, b ut inserts at most *n* wide characters, or as many as possible (up to the end of the line) if *n* is negative. **ncurses**(3X) describes the variants of these functions.

RETURN VALUE

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

In *ncurses*, they return **ERR** if

- *win* is *NULL*,
- *wstr* is *NULL*,
- the first wide character in *wstr* is a non-spacing character, or
- an internal **wins_wch**(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 **wins_nwstr** may be implemented as macros.

EXTENSIONS

ins_nwstr, **wins_nwstr**, **mvins_nwstr**, and **mvwins_nwstr**’s acceptance of negative *n* values is an *ncurses* extension.

PORTABILITY

Applications employing *ncurses* extensions should condition their use on the visibility of the **NCURSES_VERSION** preprocessor macro.

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

X/Open Curses does not specify what happens if a non-spacing character follows a control character.

Issue 4 states that the entire string is inserted if *n* is less than 1. This is probably an error, because it is inconsistent with other functions such as *waddwstr*, and differs from the SVr4 *curses* and Solaris *xcurses* implementations. Nevertheless, Issue 7 retains the language.

HISTORY

X/Open Curses Issue 4 (1995) initially specified these functions. The System V Interface Definition Version 4 (1995), specified functions named *winswstr* and *winsnwstr* (and the usual variants). These were later additions to SVr4.*x*, not appearing in the first SVr4 (1989). They differ from X/Open's later *wins_wstr* and *wins_nwstr* in that their *wstr* parameters are not *const*-qualified.

SEE ALSO

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

curses(3X), **curs_ins_wch**(3X), **curs_in_wch**(3X)