

NAME

inwstr, **winwstr**, **mvwinwstr**, **mvwinwstr**, **innwstr**, **winnwstr**, **mvinnwstr**, **mvwinnwstr** – get a wide-character string from a *curses* window

SYNOPSIS

```
#include <curses.h>

int inwstr(wchar_t * wstr);
int winwstr(WINDOW * win, wchar_t * wstr);
int mvwinwstr(int y, int x, wchar_t * wstr);
int mvwinwstr(WINDOW * win, int y, int x, wchar_t * wstr);

int innwstr(wchar_t * wstr, int n);
int winnwstr(WINDOW * win, wchar_t * wstr, int n);
int mvinnwstr(int y, int x, wchar_t * wstr, int n);
int mvwinnwstr(WINDOW * win, int y, int x, wchar_t * wstr, int n);
```

DESCRIPTION

winwstr extracts a wide-character string from a *curses* window *win*, starting at the cursor and stopping at the end of the line, and stores it in *wstr*, terminating it with a wide null character and omitting any attributes and color pair identifier that *curses* associates with each character. **winnwstr** does the same, but copies at most *n* wide characters from *win*. A negative *n* implies no limit; **winnwstr** then works like **winwstr**. **ncurses(3X)** describes the variants of these functions.

RETURN VALUE

On successful operation, these functions return the count of wide characters copied from *win* to *wstr* (not including the wide null terminator), or **ERR** upon failure. **innwstr**, **winnwstr**, **mvinnwstr**, and **mvwinnwstr** return **ERR** if *n* is insufficiently large to store a complete wide character string. (Recall that a *curses* complex character can contain multiple wide characters, some of which may be non-spacing.)

In *ncurses*, these functions return **ERR** if

- the *curses* screen has not been initialized,
- (for functions taking a *WINDOW* pointer argument) *win* is a null pointer, or
- *wstr* is a null pointer.

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 **winnwstr** may be implemented as macros.

Reading a line that overflows the array pointed to by *wstr* and its variants causes undefined results. Instead, use the *n*-infix functions with a positive *n* argument no larger than the size of the buffer backing *wstr*.

EXTENSIONS

innwstr, **winnwstr**, **mvinnwstr**, and **mvwinnwstr**’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.

Notwithstanding the foregoing, X/Open Curses Issues 4 and 7 both state that *innwstr*, *winnwstr*, *mvinnwstr*, and *mvwinnwstr* “fail ... [i]f the array is not large enough to contain any complete characters”. Strictly interpreted, this means that a caller of these functions cannot use their return values to detect truncation of a wide-character string copied from more than one character cell in *win*. *ncurses* reports any truncation with **ERR**.

X/Open Curses specifies *inwstr*, *winwstr*, *mvwinwstr*, and *mvwinwstr* as returning **OK** rather than a (wide) character count, unlike their non-wide counterparts *instr*, *winstr*, *mvinstr*, and *mvwinstr*. *ncurses* regards

this inconsistency as an error in the standard.

HISTORY

The System V Interface Definition, Version 4 (1995), specified *winwstr* and *winnwstr* (and the usual variants). These were later additions to SVr4.x, not appearing in the first SVr4 (1989). Their synopses described each function as taking an argument of type pointer-to-*char* instead of pointer-to-*wchar_t*, despite describing them as “returning the string of *wchar_t* in *str*”. Presumably this was an error in the System V Interface Definition.

SEE ALSO

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

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