

**NAME**

**add\_wchstr**, **add\_wchnstr**, **wadd\_wchstr**, **wadd\_wchnstr**, **mvadd\_wchstr**, **mvadd\_wchnstr**, **mvwadd\_wchstr**, **mvwadd\_wchnstr** - add an array of complex characters (and attributes) to a curses window

**SYNOPSIS**

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

```
int add_wchstr(const cchar_t *wchstr);
```

```
int add_wchnstr(const cchar_t *wchstr, int n);
```

```
int wadd_wchstr(WINDOW *win, const cchar_t *wchstr);
```

```
int wadd_wchnstr(WINDOW *win, const cchar_t *wchstr, int n);
```

```
int mvadd_wchstr(int y, int x, const cchar_t *wchstr);
```

```
int mvadd_wchnstr(int y, int x, const cchar_t *wchstr, int n);
```

```
int mvwadd_wchstr(WINDOW *win, int y, int x, const cchar_t *wchstr);
```

```
int mvwadd_wchnstr(WINDOW *win, int y, int x, const cchar_t *wchstr, int n);
```

**DESCRIPTION**

These functions copy the (null-terminated) array of complex characters *wchstr* into the window image structure starting at the current cursor position. The four functions with *n* as the last argument copy at most *n* elements, but no more than will fit on the line. If **n=-1** then the whole array is copied, to the maximum number of characters that will fit on the line.

The window cursor is *not* advanced. These functions work faster than **waddnstr**. On the other hand:

- ⊕ they do not perform checking (such as for the newline, backspace, or carriage return characters),
- ⊕ they do not advance the current cursor position,
- ⊕ they do not expand other control characters to ^-escapes, and
- ⊕ they truncate the string if it crosses the right margin, rather than wrapping it around to the new line.

These functions end successfully on encountering a null *cchar\_t*, or when they have filled the current line. If a complex character cannot completely fit at the end of the current line, the remaining columns are filled with the background character and rendition.

**RETURN VALUE**

`curs_add_wchstr(3X)`

`curs_add_wchstr(3X)`

All functions return the integer **ERR** upon failure and **OK** on success.

X/Open does not define any error conditions. This implementation returns an error if the window pointer is null.

Functions with a "mv" prefix first perform a cursor movement using **wmove**, and return an error if the position is outside the window, or if the window pointer is null.

## NOTES

All functions except **wadd\_wchnstr** may be macros.

## PORTABILITY

These entry points are described in the XSI Curses standard, Issue 4.

## SEE ALSO

**curses(3X)**, **curs\_addwstr(3X)**.

Comparable functions in the narrow-character (ncurses) library are described in **curs\_addchstr(3X)**.

`curs_add_wchstr(3X)`