

NAME

printw, **wprintw**, **mvprintw**, **mvwprintw**, **vprintw**, **vw_printw** – write formatted output to a *curses* window

SYNOPSIS

```
#include <curses.h>

int printw(const char *fmt, ...);
int wprintw(WINDOW *win, const char *fmt, ...);
int mvprintw(int y, int x, const char *fmt, ...);
int mvwprintw(WINDOW *win, int y, int x, const char *fmt, ...);
int vw_printw(WINDOW *win, const char *fmt, va_list varglist);
/* obsolete */
int vwprintw(WINDOW *win, const char *fmt, va_list varglist);
```

DESCRIPTION

printw, **wprintw**, **mvprintw**, and **mvwprintw** are analogous to *printf(3)*. In effect, the string that would be output by *printf(3)* is instead output as though **waddstr(3X)** were used with *win* (or **stdscr**) as its first argument.

vprintw and **vw_printw** are analogous to *vprintf(3)*, and perform a **wprintw** using a variable argument list. The third argument is a *va_list*, a pointer to a list of arguments, as defined in *stdarg.h*.

RETURN VALUE

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

In *ncurses*, failure occurs if the library cannot allocate enough memory for the buffer into which the output is formatted, or if the window pointer *win* is null.

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

NOTES

No wide character counterpart functions are defined by the “wide” *ncurses* configuration nor by any standard. To format and write a wide-character string to a *curses* window, consider using *swprintf(3)* and **waddwstr(3X)** or similar.

PORTABILITY

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

ncurses defines **vw_printw** and **vwprintw** identically to support legacy applications. However, the latter is obsolete.

- X/Open Curses Issue 4 Version 2 (1996), marked **vwprintw** as requiring *varargs.h* and “TO BE WITHDRAWN”, and specified **vw_printw** using the *stdarg.h* interface.
- X/Open Curses Issue 5, Draft 2 (December 2007) marked **vwprintw** (along with **vwscanw** and the *termcap* interface) as withdrawn. After incorporating review comments, this became X/Open Curses Issue 7 (2009).
- *ncurses* provides **vwprintw**, but marks it as deprecated.

HISTORY

4BSD (1980) introduced *wprintw* and its variants. It implemented all as functions, not macros; this initial distribution of *curses* preceded the ANSI C standard of 1989, prior to which a variadic macro facility was not widely available in the language. *printw* went unused in Berkeley distributions until 4.1cBSD (1983), which employed it in games. 4BSD’s *wprintw* did not use *varargs.h*, which had been available since Seventh Edition Unix (1979). In 1991 (a couple of years after SVr4 was generally available, and after the C standard was published), other developers updated the library, using *stdarg.h* internally in 4.4BSD *curses*. Even with this improvement, BSD *curses* did not use function prototypes (nor even declare functions) in *curses.h* until 1992.

4BSD documented *printw* and *wprintw* tersely as “printf on *stdscr*” and “printf on *win*”, respectively.

SVr3 summarized the functions in three lines, asserting that they were analogous to *printf*(3) and explaining that the string that *printf*(3) would write to the standard output stream would instead be output using *waddstr* to the given window.

SVr3 added *vwprintw*, describing its third parameter as a *va_list*, defined in *varargs.h*, and referred the reader to the manual pages for *varargs* and *vprintf* for detailed descriptions.

SVr4 (1989) introduced no new variations of *printw*, but provided for using either *varargs.h* or *stdarg.h* to define the *va_list* type.

X/Open Curses Issue 4 (1995), defined *vw_printw* to replace *vwprintw*, stating that its *va_list* type is defined in *stdarg.h*.

SEE ALSO

curses(3X), curs_addstr(3X), curs_scanw(3X), printf(3), vprintf(3)