NAME

printw, **wprintw**, **mvprintw**, **mvwprintw**, **vwprintw**, **vw_printw** - print formatted output in **curses** windows

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

The **printw**, **wprintw**, **mvprintw** and **mvwprintw** routines are analogous to **printf** [see **printf**(3)]. In effect, the string that would be output by **printf** is output instead as though **waddstr** were used on the given window.

The **vwprintw** and **vw_printw** routines are analogous to **vprintf** [see **printf**(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

Routines that return an integer return **ERR** upon failure and **OK** (SVr4 only specifies "an integer value other than **ERR**") upon successful completion.

X/Open defines no error conditions. In this implementation, an error may be returned if it cannot allocate enough memory for the buffer used to format the results. It will return 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.

HISTORY

While **printw** was implemented in 4BSD, it was unused until 4.2BSD (which used it in games). That early version of curses was before the ANSI C standard. It did not use <varargs.h>, though that was available. 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 (or even declare functions) in the <curses.h> header until 1992.

SVr2 documented **printw**, **wprintw** tersely as "printf on *stdscr*" and tersely as "printf on *win*", respectively.

SVr3 added **mvprintw**, and **mvwprintw**, with a three-line summary saying that they were analogous to **printf**(3), explaining that the string which would be output from **printf**(3) would instead be output using **waddstr** on the given window. SVr3 also added **vwprintw**, saying that the third parameter is a **va_list**, defined in <varargs.h>, and referring the reader to the manual pages for *varargs* and *vprintf* for detailed descriptions.

SVr4 added no new variations of **printw**, but provided for using <varargs.h> or <stdarg.h> to define the **va_list** type.

X/Open Curses added **vw_printw** to replace **vwprintw**, stating that its **va_list** definition requires <stdarg.h>.

PORTABILITY

In this implementation, **vw_printw** and **vwprintw** are equivalent, to support legacy applications. However, the latter (**vwprintw**) is obsolete:

- The XSI Curses standard, Issue 4 described these functions. The function **vwprintw** is marked TO BE WITHDRAWN, and is to be replaced by a function **vw_printw** using the **<stdarg.h>** interface.

- The Single Unix Specification, Version 2 states that **vw_printw** is preferred to **vwprintw** since the latter requires including **<varargs.h>**, which cannot be used in the same file as **<stdarg.h>**. This implementation uses **<stdarg.h>** for both, because that header is included in **<curses.h>**.

- X/Open Curses, Issue 5 (December 2007) marked **vwprintw** (along with **vwscanw** and the termcap interface) as withdrawn.

SEE ALSO

curses(3X), **curs_addstr**(3X), **curs_scanw**(3X), **curs_termcap**(3X), **printf**(3), **vprintf**(3).