NAME

vlfmt – display error message in standard format and pass to logging and monitoring services

MT-LEVEL

MT-safe

SYNOPSIS

#include <stdarg.h>
#include <pfmt.h>

int vlfmt(FILE *∗stream*, **long** *flags*, **char** *∗format*, **va_list** *ap*);

DESCRIPTION

vlfmt() is the same as **lfmt()** except that instead of being called with a variable number of arguments, it is called with an argument list as defined by the **<stdarg.h>** header file.

The **<stdarg.h>** header file defines the type **va_list** and a set of macros for advancing through a list of arguments whose number and types may vary. The argument *ap* to **vlfmt()** is of type **va_list**. This argument is used with the **<stdarg.h>** header file macros **va_start()**, **va_arg()** and **va_end()**.
[see **va_start()**, **va_arg()**, and **va_end()** in **stdarg**(5)]. The EXAMPLE section below shows their use with **vlfmt()**.

The macro **va_alist** is used as the parameter list in a function definition as in the function called **errlog()** in the example below. The macro **va_start(ap,)**, where *ap* is of type **va_list**, must be called before any attempt to traverse and access unnamed arguments. Calls to **va_arg(***ap*, *atype***)** traverse the argument list. Each execution of **va_arg()** expands to an expression with the value and type of the next argument in the list *ap*, which is the same object initialized by **va_start**. The argument *atype* is the type that the returned argument is expected to be. The **va_end(***ap***)** macro must be invoked when all desired arguments have been accessed. (The argument list in *ap* can be traversed again if **va_start()** is called again after **va_end()**.) In the example below, **va_arg()** is executed first to retrieve the format string passed to **errlog()**. The remainting **errlog()** arguments, *arg1*, *arg2*, ..., are given to *vlfmt()* in the argument *ap*.

RETURN VALUE

Upon success, **vlfmt()** returns the number of bytes transmitted. Upon failure, it returns a negative value:

–1 write error to *stream*.

–2 cannot log and/or display at console.

EXAMPLE

The following demonstrates how **vlfmt()** could be used to write an **errlog()** routine:

```
#include <pfmt.h>
#include <stdarg.h>
/*
 *  errlog should be called like
 *       errlog(log_info, format, arg1, ...);
 */
void errlog(long log_info, ...)
```

```
{
    va_list ap;
    char *format;

    va_start(ap, );
    format = va_arg(ap, char *);
    (void) vlfmt(stderr, log_info | MM_ERROR, format, ap);
    va_end(ap);
    (void) abort();
}
```

NOTES Since **vlfmt()** uses **gettxt**(3C), it is recommended that **vlfmt()** not be used.

SEE ALSO **gettxt**(3C), **lfmt**(3C), **stdarg**(5)