NAME

pfmt – display error message in standard format

SYNOPSIS

#include <pfmt.h>
int pfmt(FILE **stream*, **long** *flags*, **char** **format*, **... /*** *arg* ***/);**

MT-LEVEL

MT-safe

DESCRIPTION

pfmt() retrieves a format string from a locale-specific message database (unless **MM_NOGET** is specified) and uses it for **printf()** style formatting of *args*. The output is displayed on *stream*.

pfmt() encapsulates the output in the standard error message format (unless **MM_NOSTD** is specified, in which case the output is simply **printf()** like).

If the **printf()** format string is to be retrieved from a message database, the *format* argument must have the following structure:

<*catalog*>**:**<*msgnum*>**:**<*defmsg*>.

If **MM_NOGET** is specified, only the <*defmsg*> part must be specified.

<*catalog*> is used to indicate the message database that contains the localized version of the format string. <*catalog*> must be limited to 14 characters. These characters must be selected from a set of all characters values, excluding **\0** (null) and the ASCII codes for **/** (slash) and **:** (colon).

<*msgnum*> is a positive number that indicates the index of the string into the message database.

If the catalog does not exist in the locale (specified by the last call to **setlocale()** using the **LC_ALL** or **LC_MESSAGES** categories), or if the message number is out of bound, **pfmt()** will attempt to retrieve the message from the C locale. If this second retrieval fails, **pfmt()** uses the <*defmsg*> part of the *format* argument.

If <*catalog*> is omitted, **pfmt()** will attempt to retrieve the string from the default catalog specified by the last call to **setcat()**. In this case, the *format* argument has the following structure:

:<*msgnum*>**:**<*defmsg*>.

pfmt() will output **Message not found!!\n** as format string if <*catalog*> is not a valid catalog name, if no catalog is specified (either explicitely or via **setcat()**), if <*msgnum*> is not a valid number, or if no message could be retrieved from the message databases, and <*defmsg*> was omitted.

The *flags* determine the type of output (i.e. whether the *format* should be interpreted as is or encapsulated in the standard message format), and the access to message catalogs to retrieve a localized version of *format*.

The *flags* are composed of several groups, and can take the following values (one from each group): *Output format control*

MM_NOSTD Do not use the standard message format, interpret *format* as a **printf()** *format*. Only *catalog access control flags* should be specified if **MM_NOSTD** is used; all other flags will be ignored

MM_STD Output using the standard message format (default, value 0).

Catalog access control

MM_NOGET Do not retrieve a localized version of *format*. In this case, only the **<***defmsg***>** part of the *format* is specified.

MM_GET Retrieve a localized version of *format*, from the **<***catalog***>**, using **<***msgid***>** as the index and **<***defmsg***>** as the default message (default, value 0).

Severity (standard message format only)

MM_HALT generates a localized version of **HALT**, but does not halt the machine.

MM_ERROR generates a localized version of **ERROR** (default, value 0).

MM_WARNING
generates a localized version of **WARNING**.

MM_INFO generates a localized version of **INFO**.

Additional severities can be defined. Add-on severities can be defined with number-string pairs with numeric values from the range [5-255], using **addsev()**. The numeric value ORed with other *flags* will generate the specified severity.

If the severity is not defined, **pfmt()** used the string **SEV=***N* where *N* is replaced by the integer severity value passed in *flags*.

Multiple severities passed inf *flags* will not be detected as an error. Any combination of severities will be summed and the numeric value will cause the display of either a severity string (if defined) or the string **SEV=***N* (if undefined).

Action

MM_ACTION specifies an action message. Any severity value is superseded and replaced by a localized version of **TO FIX**.

STANDARD ERROR MESSAGE FORMAT

pfmt() displays error messages in the following format:
*label***:** *severity***:** *text*

If no *label* was defined by a call to **setlabel()**, the message is displayed in the format:
*severity***:** *text*

If **pfmt()** is called twice to display an error message and a helpful *action* or recovery message, the output can look like:

label: *severity*: *text*
label: **TO FIX:** *text*

RETURN VALUE

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

-1 write error to *stream*.

EXAMPLES

Example 1:

```
setlabel("UX:test");
pfmt(stderr, MM_ERROR, "test:2:Cannot open file: %s\n", strerror(errno));
```

displays the message:

```
UX:test: ERROR: Cannot open file: No such file or directory
```

Example 2:

```
setlabel("UX:test");
setcat("test");
pfmt(stderr, MM_ERROR, ":10:Syntax error\n");
pfmt(stderr, MM_ACTION, "55:Usage ...\n");
```

displays the message

```
UX:test: ERROR: Syntax error
UX:test: TO FIX: Usage ...
```

NOTES

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

SEE ALSO

addsev(3C), **gettxt**(3C), **lfmt**(3C), **printf**(3S), **setcat**(3C), **setlabel**(3C), **setlocale**(3C), **environ**(5)