NAME

fmtmsg - display a detailed diagnostic message

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <fmtmsg.h>

int
fmtmsg(*long classification, const char *label, int severity, const char *text, const char *action, const char *tag*);

DESCRIPTION

The **fmtmsg**() function displays a detailed diagnostic message, based on the supplied arguments, to stderr and/or the system console.

The *classification* argument is the bitwise inclusive OR of zero or one of the manifest constants from each of the classification groups below. The Output classification group is an exception since both MM_PRINT and MM_CONSOLE may be specified.

Output

MM_PRINT Output should take place on stderr.

MM_CONSOLE Output should take place on the system console.

Source of Condition (Major)

MM_HARD The source of the condition is hardware related.

MM_SOFT The source of the condition is software related.

MM_FIRM The source of the condition is firmware related.

Source of Condition (Minor)

MM_APPL The condition was detected at the application level.

MM_UTIL The condition was detected at the utility level.

MM_OPSYS The condition was detected at the operating system level.

Status

MM_RECOVER
The application can recover from the condition.

MM_NRECOV The application is unable to recover from the condition.

Alternatively, the MM_NULLMC manifest constant may be used to specify no classification.

The *label* argument indicates the source of the message. It is made up of two fields separated by a colon (':'). The first field can be up to 10 bytes, and the second field can be up to 14 bytes. The MM_NULLLBL manifest constant may be used to specify no label.

The *severity* argument identifies the importance of the condition. One of the following manifest constants should be used for this argument.

MM_HALT The application has confronted a serious fault and is halting.

MM_ERROR The application has detected a fault.

MM_WARNING The application has detected an unusual condition, that could be indicative of a problem.

MM_INFO The application is providing information about a non-error condition.

MM_NOSEV No severity level supplied.

The *text* argument details the error condition that caused the message. There is no limit on the size of this character string. The MM_NULLTXT manifest constant may be used to specify no text.

The *action* argument details how the error-recovery process should begin. Upon output, **fmtmsg**() will prefix "TO FIX:" to the beginning of the *action* argument. The MM_NULLACT manifest constant may be used to specify no action.

The *tag* argument should reference online documentation for the message. This usually includes the *label* and a unique identifying number. An example tag is "BSD:ls:168". The MM_NULLTAG manifest constant may be used to specify no tag.

RETURN VALUES

The **fmtmsg**() function returns MM_OK upon success, MM_NOMSG to indicate output to stderr failed, MM_NOCON to indicate output to the system console failed, or MM_NOTOK to indicate output to stderr and the system console failed.

ENVIRONMENT

The MSGVERB (message verbosity) environment variable specifies which arguments to **fmtmsg**() will be output to stderr, and in which order. MSGVERB should be a colon (‘:’) separated list of identifiers. Valid identifiers include: label, severity, text, action, and tag. If invalid identifiers are specified or incorrectly separated, the default message verbosity and ordering will be used. The default ordering is equivalent to a MSGVERB with a value of "label:severity:text:action:tag".

EXAMPLES

The code:

```
fmtmsg(MM_UTIL | MM_PRINT, "BSD:ls", MM_ERROR,
   "illegal option -- z", "refer to manual", "BSD:ls:001");
```

will output:

```
BSD:ls: ERROR: illegal option -- z
TO FIX: refer to manual BSD:ls:001
```

to stderr.

The same code, with MSGVERB set to "text:severity:action:tag", produces:

```
illegal option -- z: ERROR
TO FIX: refer to manual BSD:ls:001
```

SEE ALSO

err(3), exit(3), strerror(3)

STANDARDS

The **fmtmsg**() function conforms to IEEE Std 1003.1-2001 ("POSIX.1").

HISTORY

The **fmtmsg**() function first appeared in FreeBSD 5.0.

BUGS

Specifying MM_NULLMC for the *classification* argument makes little sense, since without an output specified, **fmtmsg**() is unable to do anything useful.

In order for **fmtmsg**() to output to the system console, the effective user must have appropriate permission to write to */dev/console*. This means that on most systems **fmtmsg**() will return MM_NOCON unless the effective user is root.