Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'fmtmsg.3p' command

$ man fmtmsg.3p

FMTMSG(3P) POSIX Programmer's Manual FMTMSG(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

fmtmsg ? display a message in the specified format on standard error and/or a system console

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 shall display messages in a specified format in? stead of the traditional printf() function.

Based on a message's classification component, fmtmsg() shall write a formatted message either to standard error, to the console, or to both.

A formatted message consists of up to five components as defined below.

The component classification is not part of a message displayed to the user, but defines the source of the message and directs the display of the formatted message.

classification

Contains the sum of identifying values constructed from the constants defined below. Any one identifier from a subclass may be used in combination with a single identifier from a different subclass. Two or more identifiers from the same subclass should not be used together, with the exception of identifiers from the display subclass. (Both display sub? class identifiers may be used so that messages can be dis? played to both standard error and the system console.)

Major Classifications

Identifies the source of the condition. Identifiers are: MM_HARD (hardware), MM_SOFT (software), and MM_FIRM (firmware).

Message Source Subclassifications

Identifies the type of software in which the problem is detected. Identifiers are: MM_APPL (application), MM_UTIL (utility), and MM_OPSYS (operating system).

Display Subclassifications

Indicates where the message is to be displayed. Iden? tifiers are: MM_PRINT to display the message on the standard error stream, MM_CONSOLE to display the mes? sage on the system console. One or both identifiers may be used.

Status Subclassifications

Indicates whether the application can recover from the condition. Identifiers are: MM_RECOVER (recover? able) and MM_NRECOV (non-recoverable).

An additional identifier, MM_NULLMC, indicates that no classification component is supplied for the message.

label Identifies the source of the message. The format is two fields separated by a <colon>. The first field is up to 10 bytes, the second is up to 14 bytes.

severity Indicates the seriousness of the condition. Identifiers for the levels of severity are:

MM_HALT Indicates that the application has encountered a severe fault and is halting. Produces the string "HALT".

MM_ERROR Indicates that the application has detected a fault. Produces the string "ERROR".

MM_WARNING Indicates a condition that is out of the ordi? nary, that might be a problem, and should be watched. Produces the string "WARNING".

MM_INFO Provides information about a condition that is not in error. Produces the string "INFO".

MM_NOSEV Indicates that no severity level is supplied for the message.

text Describes the error condition that produced the message. The character string is not limited to a specific size. If the character string is empty, then the text produced is unspecified.

action Describes the first step to be taken in the error-recovery process. The fmtmsg() function precedes the action string with the prefix: "TOFIX:". The action string is not lim? ited to a specific size.

tag An identifier that references on-line documentation for the message. Suggested usage is that tag includes the label and a unique identifying number. A sample tag is "XSI:cat:146".

The MSGVERB environment variable (for message verbosity) shall deter? mine for fmtmsg() which message components it is to select when writing messages to standard error. The value of MSGVERB shall be a <colon>-separated list of optional keywords. Valid keywords are: label, severity, text, action, and tag. If MSGVERB contains a keyword for a component and the component's value is not the component's null value, fmtmsg() shall include that component in the message when writing the message to standard error. If MSGVERB does not include a keyword for a message component, that component shall not be included in the display

of the message. The keywords may appear in any order. If MSGVERB is not defined, if its value is the null string, if its value is not of the correct format, or if it contains keywords other than the valid ones listed above, fmtmsg() shall select all components.

MSGVERB shall determine which components are selected for display to standard error. All message components shall be included in console messages.

RETURN VALUE

The fmtmsg() function shall return one of the following values:

MM_OK The function succeeded.

MM_NOTOK The function failed completely.

MM_NOMSG The function was unable to generate a message on standard error, but otherwise succeeded.

MM_NOCON The function was unable to generate a console message, but otherwise succeeded.

ERRORS

None.

The following sections are informative.

EXAMPLES

1. The following example of fmtmsg():

   fmtmsg(MM_PRINT, "XSI:cat", MM_ERROR, "illegal option", "refer to cat in user's reference manual", "XSI:cat:001")

   produces a complete message in the specified message format:

   XSI:cat: ERROR: illegal option
   TO FIX: refer to cat in user's reference manual XSI:cat:001

2. When the environment variable MSGVERB is set as follows:

   MSGVERB=severity:text:action

   and Example 1 is used, fmtmsg() produces:

   ERROR: illegal option
   TO FIX: refer to cat in user's reference manual

APPLICATION USAGE

One or more message components may be systematically omitted from mes? sages generated by an application by using the null value of the argu?

ment for that component.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

fprintf()

The Base Definitions volume of POSIX.1?2017, <fmtmsg.h>

COPYRIGHT