

**NAME**

addseverity – introduce new severity classes

**SYNOPSIS**

```
#include <fmtmsg.h>
```

```
int addseverity(int severity, const char *s);
```

Feature Test Macro Requirements for glibc (see [feature\\_test\\_macros\(7\)](#)):

**addseverity()**: Since glibc 2.19: `_DEFAULT_SOURCE` Glibc 2.19 and earlier: `_SVID_SOURCE`

**DESCRIPTION**

This function allows the introduction of new severity classes which can be addressed by the *severity* argument of the [fmtmsg\(3\)](#) function. By default, that function knows only how to print messages for severity 0-4 (with strings (none), HALT, ERROR, WARNING, INFO). This call attaches the given string *s* to the given value *severity*. If *s* is NULL, the severity class with the numeric value *severity* is removed. It is not possible to overwrite or remove one of the default severity classes. The severity value must be nonnegative.

**RETURN VALUE**

Upon success, the value `MM_OK` is returned. Upon error, the return value is `MM_NOTOK`. Possible errors include: out of memory, attempt to remove a nonexistent or default severity class.

**VERSIONS**

**addseverity()** is provided in glibc since version 2.1.

**ATTRIBUTES**

For an explanation of the terms used in this section, see [attributes\(7\)](#).

Interface	Attribute	Value
<b>addseverity()</b>	Thread safety	MT-Safe

**CONFORMING TO**

This function is not specified in the X/Open Portability Guide although the [fmtmsg\(3\)](#) function is. It is available on System V systems.

**NOTES**

New severity classes can also be added by setting the environment variable `SEV_LEVEL`.

**SEE ALSO**

[fmtmsg\(3\)](#)

**COLOPHON**

This page is part of release 4.16 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.