```
  1: /*    Copyright (c) 1993 UNIX System Laboratories, Inc.    */
  2: /*    (a wholly-owned subsidiary of Novell, Inc.).         */
  3: /*    All Rights Reserved.                                 */
  4:
  5: /*    THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF UNIX SYSTEM        */
  6: /*    LABORATORIES, INC. (A WHOLLY-OWNED SUBSIDIARY OF NOVELL, INC.).   */
  7: /*    The copyright notice above does not evidence any actual or        */
  8: /*    intended publication of such source code.                         */
  9:
 10: #ifndef _FMTMSG_H
 11: #define _FMTMSG_H
...
 68:
 69: /*
 70:  * Definitions of type classifications:
 71:  *    MM_HARD        Hardware
 72:  *    MM_SOFT        Software
 73:  *    MM_FIRM        Firmware
 74:  */
 75:
 76: #define    MM_HARD        0x00000001L
 77: #define    MM_SOFT        0x00000002L
 78: #define    MM_FIRM        0x00000004L
 79:
 80:
 81: /*
 82:  * Definitions of recoverability subclassification
 83:  *    MM_RECOVER    Recoverable
 84:  *    MM_NRECOV    Non-recoverable
 85:  */
 86:
 87: #define    MM_RECOVER    0x00000100L
 88: #define    MM_NRECOV    0x00000200L
 89:
 90:
 91: /*
 92:  * Definitions of source subclassification
 93:  *    MM_APPL        Application
 94:  *    MM_UTIL        Utility
 95:  *    MM_OPSYS    Kernel
 96:  */
 97:
 98: #define    MM_APPL        0x00000008L
 99: #define MM_UTIL        0x00000010L
100: #define    MM_OPSYS    0x00000020L
101:
102:
103: /*
104:  * Definitions for the action to take with the message:
105:  *    MM_PRINT    Write to the standard error stream
106:  *    MM_CONSOLE    Treat the message as a console message
107:  */
108:
109: #define    MM_PRINT    0x00000040L
110: #define    MM_CONSOLE    0x00000080L
111:
112: /*
113:  * Constants for severity values
114:  *
115:  *    SEV_LEVEL    Names the env variable that defines severities
116:  *
117:  *    MM_NOSEV    Message has no severity
118:  *    MM_HALT        Message describes a severe error condition
119:  *    MM_ERROR    Message describes an error condition
120:  *    MM_WARNING    Message tells of probable error condition
121:  *    MM_INFO        Message informs, not in error
122:  */
123:
124: #define    SEV_LEVEL    "SEV_LEVEL"
125: #define MM_NOSEV    0
126: #define    MM_HALT        1
127: #define    MM_ERROR    2
128: #define    MM_WARNING    3
129: #define    MM_INFO        4
130:
131: /*
132:  * Null values for message components
133:  *    MM_NULLLBL        Null value for the label-component
```

```
  1: /* Message display handling.
  2:    Copyright (C) 1997, 1999, 2000 Free Software Foundation, Inc.
  3:    This file is part of the GNU C Library.
  4:
  5:    The GNU C Library is free software; you can redistribute it and/or
  6:    modify it under the terms of the GNU Lesser General Public
  7:    License as published by the Free Software Foundation; either
  8:    version 2.1 of the License, or (at your option) any later version.
  9:
 10:    The GNU C Library is distributed in the hope that it will be useful,
 11:    but WITHOUT ANY WARRANTY; without even the implied warranty of
 12:    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 13:    Lesser General Public License for more details.
 14:
 15:    You should have received a copy of the GNU Lesser General Public
 16:    License along with the GNU C Library; if not, write to the Free
 17:    Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
 18:    02111-1307 USA.  */
 19:
 20: #ifndef __FMTMSG_H
 21: #define __FMTMSG_H    1
 22:
 23: #include <features.h>
 24:
 25:
 26: __BEGIN_DECLS
 27:
 28: /* Values to control `fmtmsg' function.  */
 29: enum
 30: {
 31:   MM_HARD = 0x001,    /* Source of the condition is hardware.  */
 32: #define MM_HARD MM_HARD
 33:   MM_SOFT = 0x002,    /* Source of the condition is software.  */
 34: #define MM_SOFT MM_SOFT
 35:   MM_FIRM = 0x004,    /* Source of the condition is firmware.  */
 36: #define MM_FIRM MM_FIRM
 37:   MM_APPL = 0x008,    /* Condition detected by application.  */
 38: #define MM_APPL MM_APPL
 39:   MM_UTIL = 0x010,    /* Condition detected by utility.  */
 40: #define MM_UTIL MM_UTIL
 41:   MM_OPSYS = 0x020,    /* Condition detected by operating system.  */
 42: #define MM_OPSYS MM_OPSYS
 43:   MM_RECOVER = 0x040,    /* Recoverable error.  */
 44: #define MM_RECOVER MM_RECOVER
 45:   MM_NRECOV = 0x080,    /* Non-recoverable error.  */
 46: #define MM_NRECOV MM_NRECOV
 47:   MM_PRINT = 0x100,    /* Display message in standard error.  */
 48: #define MM_PRINT MM_PRINT
 49:   MM_CONSOLE = 0x200    /* Display message on system console.  */
 50: #define MM_CONSOLE MM_CONSOLE
 51: };
 52:
 53: /* Values to be for SEVERITY parameter of `fmtmsg'.  */
 54: enum
 55: {
 56:   MM_NOSEV = 0,        /* No severity level provided for the message.  */ [#m11]
 57: #define MM_NOSEV MM_NOSEV
 58:   MM_HALT,        /* Error causing application to halt.  */
 59: #define MM_HALT MM_HALT
 60:   MM_ERROR,        /* Application has encountered a non-fatal fault.  */
 61: #define MM_ERROR MM_ERROR
 62:   MM_WARNING,        /* Application has detected unusual non-error
 63:                 condition.  */
 64: #define MM_WARNING MM_WARNING
 65:   MM_INFO        /* Informative message.  */
 66: #define MM_INFO MM_INFO
 67: };
 68:
 69:
 70: /* Macros which can be used as null values for the arguments of `fmtmsg'.  */
 71: #define MM_NULLLBL    ((char *) 0)
 72: #define MM_NULLSEV    0
 73: #define MM_NULLMC    ((long int) 0)
 74: #define MM_NULLTXT    ((char *) 0)
 75: #define MM_NULLACT    ((char *) 0)
 76: #define MM_NULLTAG    ((char *) 0)
 77:
 78:
```


(For readability, not all matches are indicated.)

```
134:  *    MM_NULLSEV        Null value for the severity-component
135:  *    MM_NULLMC        Null value for the classification-component
136:  *    MM_NULLTXT        Null value for the text-component
137:  *    MM_NULLACT        Null value for the action-component
138:  *    MM_NULLTAG        Null value for the tag-component
139:  */
140:
141: #define    MM_NULLLBL    ((char *) NULL)
142: #define    MM_NULLSEV    MM_NOSEV
143: #define    MM_NULLMC    MM_NULL
144: #define    MM_NULLTXT    ((char *) NULL)
145: #define    MM_NULLACT    ((char *) NULL)
146: #define    MM_NULLTAG    ((char *) NULL)
147:
148: /*
149:  * Values returned by fmtmsg()
150:  *
151:  *    MM_NOTOK    None of the requested messages were generated
152:  *    MM_NOMSG    No message was written to stderr
153:  *    MM_NOCON    No console message was generated
154:  */
155:
156: #define    MM_NOTOK    -1
157: #define    MM_OK        0x00
158: #define    MM_NOMSG    0x01
159: #define MM_NOCON    0x04
160:
161:
162: /* Function definition */
163:
164: #if defined(__STDC__)
165: int    fmtmsg(long, const char *, int, const char *, const char *, const char *);
166: int    addseverity(int, const char *);
167:
168: #else
169: int    fmtmsg();
170: int    addseverity();
171: #endif
172:
173: #endif /* _FMTMSG_H */
```



```
 79: /* Possible return values of `fmtmsg'.  */
 80: enum
 81: {
 82:   MM_NOTOK = -1,
 83: #define MM_NOTOK MM_NOTOK
 84:   MM_OK = 0,
 85: #define MM_OK MM_OK
 86:   MM_NOMSG = 1,
 87: #define MM_NOMSG MM_NOMSG
 88:   MM_NOCON = 4
 89: #define MM_NOCON MM_NOCON
 90: };
 91:
 92:
 93: /* Print message with given CLASSIFICATION, LABEL, SEVERITY, TEXT, ACTION
 94:    and TAG to console or standard error.  */
 95: extern int fmtmsg (long int __classification, __const char *__label,
 96:            int __severity, __const char *__text,
 97:            __const char *__action, __const char *__tag) __THROW;
 98:
 99: #ifdef __USE_SVID
100: /* Add or remove severity level.  */
101: extern int addseverity (int __severity, __const char *__string) __THROW;
102: #endif
103:
104: __END_DECLS
105:
106: #endif /* fmtmsg.h */
```


(For readability, not all matches are indicated.)