

```

*****
      8231 Thu Oct 23 20:00:13 2014
new/usr/src/man/man3c/mq_send.3c
5258 Incorrect range specified in mq_send(3c) manual
*****
1  \" te
2  .\" Copyright (c) 2014, Ryan Zezeski.
3  .\" Copyright (c) 2008, Sun Microsystems, Inc. All Rights Reserved.
4  .\" Copyright 1989 AT&T
5  .\" Portions Copyright (c) 1992, X/Open Company Limited. All Rights Reserved.
6  .\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
7  .\" http://www.opengroup.org/bookstore/.
8  .\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
9  .\" This notice shall appear on any product containing this material.
10 .\" The contents of this file are subject to the terms of the Common Development
11 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
12 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
13 .TH MQ_SEND 3C "Oct 23, 2014"
12 .TH MQ_SEND 3C "Feb 5, 2008"
14 .SH NAME
15 mq_send, mq_timedsend, mq_reltimedsend_np \- send a message to a message queue
16 .SH SYNOPSIS
17 .LP
18 .nf
19 #include <mqqueue.h>

21 \fBint\fR \fBmq_send\fR(\fBmqd_t\fR \fBmqdes\fR, \fBconst char *\fR \fBmsg_ptr\fR
22     \fBbunsignd\fR \fBprio\fR);
23 .fi

25 .LP
26 .nf
27 #include <mqqueue.h>
28 #include <time.h>

30 \fBint\fR \fBmq_timedsend\fR(\fBmqd_t\fR \fBmqdes\fR, \fBconst char *\fR \fBmsg_ptr\fR
31     \fBbsize_t\fR \fBlen\fR, \fBbunsignd\fR \fBprio\fR,
32     \fBconst struct timespec *restrict\fR \fBtspec\fR);
33 .fi

35 .LP
36 .nf
37 \fBint\fR \fBmq_reltimedsend_np\fR(\fBmqd_t\fR \fBmqdes\fR, \fBconst char *\fR \fBmsg_ptr\fR
38     \fBbsize_t\fR \fBlen\fR, \fBbunsignd\fR \fBprio\fR,
39     \fBconst struct timespec *restrict\fR \fBirel_timeout\fR);
40 .fi

42 .SH DESCRIPTION
43 .sp
44 .LP
45 The \fBmq_send()\fR function adds the message pointed to by the argument
46 \fBmsg_ptr\fR to the message queue specified by \fBmqdes\fR. The \fBlen\fR
47 argument specifies the length of the message in bytes pointed to by
48 \fBmsg_ptr\fR. The value of \fBlen\fR is less than or equal to the
49 \fBmq_msgsize\fR attribute of the message queue, or \fBmq_send()\fR fails.
50 .sp
51 .LP
52 If the specified message queue is not full, \fBmq_send()\fR behaves as
53 if the message is inserted into the message queue at the position
54 indicated by the \fBprio\fR argument. A message with a larger
55 numeric value of \fBprio\fR is inserted before messages with lower
56 values of \fBprio\fR. A message will be inserted after other
57 messages in the queue, if any, with equal \fBprio\fR. The value of
58 \fBprio\fR must range from zero to \fBmq_max_priority - 1\fR.
59 If the specified message queue is not full, \fBmq_send()\fR behaves as if the
60 message is inserted into the message queue at the position indicated by the

```

```

61 \fBprio\fR argument. A message with a larger numeric value of
62 \fBprio\fR is inserted before messages with lower values of \fBprio\fR.
63 A message will be inserted after other messages in the queue, if any, with
64 equal \fBprio\fR. The value of \fBprio\fR must be greater than zero and
65 less than or equal to \fBmq_max_priority\fR.
66 .sp
67 .LP
68 If the specified message queue is full and \fBmq_nonblock\fR is not set in the
69 message queue description associated with \fBmqdes\fR (see \fBmq_open(3C)\fR
70 and \fBmq_setattr(3C)\fR), \fBmq_send()\fR blocks until space becomes available
71 to enqueue the message, or until \fBmq_send()\fR is interrupted by a signal. If
72 more than one thread is waiting to send when space becomes available in the
73 message queue, then the thread of the highest priority which has been waiting
74 the longest is unblocked to send its message. Otherwise, it is unspecified
75 which waiting thread is unblocked. If the specified message queue is full and
76 \fBmq_nonblock\fR is set in the message queue description associated with
77 \fBmqdes\fR, the message is not queued and \fBmq_send()\fR returns an error.
78 .sp
79 .LP
80 The \fBmq_timedsend()\fR function adds a message to the message queue specified
81 by \fBmqdes\fR in the manner defined for the \fBmq_send()\fR function. However,
82 if the specified message queue is full and \fBmq_nonblock\fR is not set in the
83 message queue description associated with \fBmqdes\fR, the wait for sufficient
84 room in the queue is terminated when the specified timeout expires. If
85 \fBmq_nonblock\fR is set in the message queue description, this function is
86 equivalent to \fBmq_send()\fR.
87 .sp
88 .LP
89 The \fBmq_reltimedsend_np()\fR function is identical to the
90 \fBmq_timedsend()\fR function, except that the timeout is specified as a
91 relative time interval.
92 .sp
93 .LP
94 For \fBmq_timedsend()\fR, the timeout expires when the absolute time specified
95 by \fBabs_timeout\fR passes, as measured by the \fBclock_gettime()\fR clock
96 (that is, when the value of that clock equals or exceeds \fBabs_timeout\fR), or
97 if the absolute time specified by \fBabs_timeout\fR has already been passed at
98 the time of the call.
99 .sp
100 .LP
101 For \fBmq_reltimedsend_np()\fR, the timeout expires when the time interval
102 specified by \fBirel_timeout\fR passes, as measured by the \fBclock_gettime()\fR
103 clock, or if the time interval specified by \fBirel_timeout\fR is negative at
104 the time of the call.
105 .sp
106 .LP
107 The resolution of the timeout is the resolution of the \fBclock_gettime()\fR
108 clock. The \fBtimespec\fR argument is defined in the <\fBtime.h\fR> header.
109 .sp
110 .LP
111 Under no circumstance does the operation fail with a timeout if there is
112 sufficient room in the queue to add the message immediately. The validity of
113 the timeout parameter need not be checked when there is sufficient room in the
114 queue.
115 .SH RETURN VALUES
116 .sp
117 .LP
118 Upon successful completion, \fBmq_send()\fR, \fBmq_timedsend()\fR, and
119 \fBmq_reltimedsend_np()\fR return \fB0\fR. Otherwise, no message is enqueued,
120 the functions return \fB-1\fR, and \fBerrno\fR is set to indicate the error.
121 .SH ERRORS
122 .sp
123 .LP
124 The \fBmq_send()\fR, \fBmq_timedsend()\fR, and \fBmq_reltimedsend_np()\fR
125 functions will fail if:
126 .sp

```

```

120 .ne 2
121 .na
122 \fB\fBEAGAIN\fR \fR
123 .ad
124 .RS 13n
125 The \fBO_NONBLOCK\fR flag is set in the message queue description associated
126 with \fImqdes\fR, and the specified message queue is full.
127 .RE

129 .sp
130 .ne 2
131 .na
132 \fB\fBEBADF\fR \fR
133 .ad
134 .RS 13n
135 The \fImqdes\fR argument is not a valid message queue descriptor open for
136 writing.
137 .RE

139 .sp
140 .ne 2
141 .na
142 \fB\fBEINTR\fR \fR
143 .ad
144 .RS 13n
145 A signal interrupted the function call.
146 .RE

148 .sp
149 .ne 2
150 .na
151 \fB\fBEINVAL\fR \fR
152 .ad
153 .RS 13n
154 The value of \fImsg_prio\fR was outside the valid range.
155 .RE

157 .sp
158 .ne 2
159 .na
160 \fB\fBEINVAL\fR \fR
161 .ad
162 .RS 13n
163 The process or thread would have blocked, and the timeout parameter specified a
164 nanoseconds field value less than zero or greater than or equal to 1,000
165 million.
166 .RE

168 .sp
169 .ne 2
170 .na
171 \fB\fBEMSGSIZE\fR \fR
172 .ad
173 .RS 13n
174 The specified message length, \fImsg_len\fR, exceeds the message size attribute
175 of the message queue.
176 .RE

178 .sp
179 .ne 2
180 .na
181 \fB\fBETIMEDOUT\fR \fR
182 .ad
183 .RS 13n
184 The \fBO_NONBLOCK\fR flag was not set when the message queue was opened, but
185 the timeout expired before the message could be added to the queue.

```

```

186 .RE

188 .SH ATTRIBUTES
189 .sp
190 .LP
191 See \fBattributes\fR(5) for descriptions of the following attributes:
192 .sp

194 .sp
195 .TS
196 box;
197 l | l
198 l | l .
199 ATTRIBUTE TYPE ATTRIBUTE VALUE
200 -
201 Interface Stability Committed
202 -
203 MT-Level MT-Safe
204 -
205 Standard See below.
206 .TE

208 .sp
209 .LP
210 For \fBmq_send()\fR and \fBmq_timedsend()\fR, see \fBstandards\fR(5).
211 .SH SEE ALSO
212 .sp
213 .LP
214 \fBsysconf\fR(3C), \fBmqqueue.h\fR(3HEAD), \fBmq_open\fR(3C),
215 \fBmq_receive\fR(3C), \fBmq_setattr\fR(3C), \fBattributes\fR(5),
216 \fBstandards\fR(5)

```