The cl-posix-mqueue Reference Manual

POSIX message queue bindings for Common Lisp, version 0.1.1

Valeriy Litkovskyy <vlr.ltkvsk@protonmail.com>

This manual was generated automatically by Declt 3.0 "Montgomery Scott" on Sun Mar 28 15:51:40 2021 GMT+1.

Table of Contents

1 Systems

The main system appears first, followed by any subsystem dependency.

1.1 cl-posix-mqueue

Author Valeriy Litkovskyy <vlr.ltkvsk@protonmail.com>

License GPL3

Description
POSIX message queue bindings for Common Lisp

Long Description
Common Lisp bindings to POSIX message queues.

POSIX message queue is an IPC (Inter-Process Communication) method that is easy to use and quick to setup.

This library uses https://common-lisp.net/project/local-time library for timestamps.

Other dependencies are: alexandria, babel and cffi. Cffi should be able to find librt.

Version 0.1.1

Dependencies
- cffi
- alexandria
- babel
- local-time

Source [cl-posix-mqueue.asd], page 5, (file)

Component
[src], page 3, (module)

2 Modules

Modules are listed depth-first from the system components tree.

2.1 cl-posix-mqueue/src

Parent [cl-posix-mqueue], page 1, (system)

Location src/

Components

- [package.lisp], page 5, (file)
- [condition.lisp], page 5, (file)
- [lib.lisp], page 6, (file)
- [queue.lisp], page 7, (file)
- [spec.lisp], page 7, (file)
- [translation.lisp], page 8, (file)
- [types.lisp], page 8, (file)

3 Files

Files are sorted by type and then listed depth-first from the systems components trees.

3.1 Lisp

3.1.1 cl-posix-mqueue.asd

Location cl-posix-mqueue.asd

Systems [cl-posix-mqueue], page 1, (system)

3.1.2 cl-posix-mqueue/src/package.lisp

Parent [src], page 3, (module)

Location src/package.lisp

Packages [posix-mqueue], page 9,

3.1.3 cl-posix-mqueue/src/condition.lisp

Parent [src], page 3, (module)

Location src/condition.lisp

Exported Definitions

- [access-denied-on-unlink], page 21, (condition)
- [access-denied-permission], page 22, (condition)
- [access-denied-slashes], page 22, (condition)
- [bad-file-descriptor-invalid], page 22, (condition)
- [bad-file-descriptor-on-receive], page 22, (condition)
- [bad-file-descriptor-on-send], page 23, (condition)
- [file-exists], page 23, (condition)
- [file-table-overflow], page 23, (condition)
- [interrupted-system-call], page 23, (condition)
- [invalid-argument-attributes], page 24, (condition)
- [invalid-argument-name], page 24, (condition)
- [invalid-argument-on-send-receive], page 24, (condition)
- [invalid-argument-on-unlink], page 24, (condition)
- [invalid-argument-sizes], page 25, (condition)
- [message-too-long-on-receive], page 25, (condition)
- [message-too-long-on-send], page 25, (condition)
- [name-too-long], page 26, (condition)
- [no-file-or-directory-just-slash], page 26, (condition)
- [no-file-or-directory-no-create], page 26, (condition)
- [no-file-or-directory-on-unlink], page 26, (condition)
- [no-space-left-on-device], page 27, (condition)
- [out-of-memory], page 27, (condition)
- [too-many-open-files], page 27, (condition)

Internal Definitions

- [access-denied], page 32, (condition)
- [bad-file-descriptor], page 33, (condition)
- [generic], page 33, (condition)
- [invalid-argument], page 34, (condition)
- [message], page 32, (method)
- [message-too-long], page 34, (condition)
- [no-file-or-directory], page 34, (condition)
- [strerror], page 32, (method)

3.1.4 cl-posix-mqueue/src/lib.lisp

Dependencies

- [condition.lisp], page 5, (file)
- [queue.lisp], page 7, (file)
- [spec.lisp], page 7, (file)
- [types.lisp], page 8, (file)

Parent [src], page 3, (module)

Location src/lib.lisp

Exported Definitions

- [*retry-on-interrupt-p*], page 13, (special variable)
- [attributes], page 13, (function)
- [close-queue], page 13, (function)
- [default-sizes], page 14, (function)
- [open-queue], page 15, (function)
- [receive], page 17, (function)
- [receive-buffer], page 18, (function)
- [receive-displaced], page 18, (function)
- [receive-string], page 18, (function)
- [send], page 18, (function)
- [send-string], page 19, (function)
- [set-non-blocking], page 19, (function)
- [timed-receive], page 20, (function)
- [timed-receive-buffer], page 20, (function)
- [timed-receive-displaced], page 20, (function)
- [timed-receive-string], page 20, (function)
- [timed-send], page 20, (function)
- [timed-send-string], page 21, (function)
- [unlink], page 21, (function)
- [with-open-queue], page 13, (macro)

Internal Definitions

- [%receive], page 29, (macro)
- [%send], page 30, (macro)
- [random-queue-name], page 32, (function)

3.1.5 cl-posix-mqueue/src/queue.lisp

Parent [src], page 3, (module)

Location src/queue.lisp

Exported Definitions

- [attributes], page 28, (structure)
- [buffer], page 13, (function)
- [(setf buffer)], page 13, (function)
- [current-messages], page 14, (function)
- [(setf current-messages)], page 14, (function)
- [max-messages], page 14, (function)
- [(setf max-messages)], page 14, (function)
- [message-size], page 14, (function)
- [(setf message-size)], page 14, (function)
- [non-blocking-p], page 14, (function)
- [(setf non-blocking-p)], page 14, (function)
- [queue], page 28, (structure)

Internal Definitions

- [attributes-p], page 30, (function)
- [copy-attributes], page 30, (function)
- [make-attributes], page 30, (function)
- [make-queue], page 30, (function)
- [mqd], page 32, (function)
- [(setf mqd)], page 32, (function)
- [queue-p], page 32, (function)

3.1.6 cl-posix-mqueue/src/spec.lisp

Dependencies

- [queue.lisp], page 7, (file)
- [translation.lisp], page 8, (file)
- [types.lisp], page 8, (file)

Parent [src], page 3, (module)

Location src/spec.lisp

Internal Definitions

- [mq-close], page 30, (function)
- [mq-close-default], page 30, (function)
- [mq-getattr], page 30, (function)
- [mq-getattr-default], page 31, (function)
- [mq-open], page 31, (function)
- [mq-open-default], page 31, (function)
- [mq-receive], page 31, (function)
- [mq-send], page 31, (function)
- [mq-setattr], page 31, (function)

- [mq-timedreceive], page 31, (function)
- [mq-timedsend], page 31, (function)
- [mq-unlink], page 31, (function)

3.1.7 cl-posix-mqueue/src/translation.lisp

Dependencies

- [queue.lisp], page 7, (file)
- [types.lisp], page 8, (file)

Parent [src], page 3, (module)

Location src/translation.lisp

Internal Definitions

- [%var-accessor-*errno*], page 30, (function)
- [(setf %var-accessor-*errno*)], page 30, (function)
- [*errno*], page 29, (symbol macro)

3.1.8 cl-posix-mqueue/src/types.lisp

Parent [src], page 3, (module)

Location src/types.lisp

Exported Definitions

- [create-modes], page 29, (type)
- [create-modesp], page 14, (function)
- [open-flags], page 29, (type)
- [open-flagsp], page 14, (function)

Internal Definitions

- [mq-attr-tclass], page 35, (class)
- [mq-get-attr-type], page 35, (class)
- [mq-non-blocking-attr-type], page 35, (class)
- [mq-size-attr-type], page 35, (class)
- [mqd-type], page 36, (class)
- [result-type], page 36, (class)
- [timespec-tclass], page 36, (class)
- [timespec-type], page 36, (class)

4 Packages

Packages are listed by definition order.

4.1 posix-mqueue

POSIX message queue bindings.

Source [package.lisp], page 5, (file)

Use List common-lisp

Exported Definitions

- [*retry-on-interrupt-p*], page 13, (special variable)
- [access-denied-on-unlink], page 21, (condition)
- [access-denied-permission], page 22, (condition)
- [access-denied-slashes], page 22, (condition)
- [attributes], page 13, (function)
- [attributes], page 28, (structure)
- [bad-file-descriptor-invalid], page 22, (condition)
- [bad-file-descriptor-on-receive], page 22, (condition)
- [bad-file-descriptor-on-send], page 23, (condition)
- [buffer], page 13, (function)
- [(setf buffer)], page 13, (function)
- [close-queue], page 13, (function)
- [create-modes], page 29, (type)
- [create-modesp], page 14, (function)
- [current-messages], page 14, (function)
- [(setf current-messages)], page 14, (function)
- [default-sizes], page 14, (function)
- [file-exists], page 23, (condition)
- [file-table-overflow], page 23, (condition)
- [interrupted-system-call], page 23, (condition)
- [invalid-argument-attributes], page 24, (condition)
- [invalid-argument-name], page 24, (condition)
- [invalid-argument-on-send-receive], page 24, (condition)
- [invalid-argument-on-unlink], page 24, (condition)
- [invalid-argument-sizes], page 25, (condition)
- [max-messages], page 14, (function)
- [(setf max-messages)], page 14, (function)
- [message-size], page 14, (function)
- [(setf message-size)], page 14, (function)
- [message-too-long-on-receive], page 25, (condition)
- [message-too-long-on-send], page 25, (condition)
- [name-too-long], page 26, (condition)
- [no-file-or-directory-just-slash], page 26, (condition)

Internal Definitions

5 Definitions

Definitions are sorted by export status, category, package, and then by lexicographic order.

5.1 Exported definitions

5.1.1 Special variables

`*retry-on-interrupt-p*` [Special Variable]
Whether or not to retry send/receive operation on interrupt.

Package [`posix-mqueue`], page 9,

Source [`lib.lisp`], page 6, (file)

5.1.2 Macros

`with-open-queue` (*VAR NAME* **&rest** *OPTIONS*) **&body** *BODY* [Macro]
A macro that automatically closes opened queue, even when condition is signaled. For OPTIONS see OPEN-QUEUE.

Example:

(with-open-queue (mqueue "/myqueue" :open-flags '(:read-write :create)) (do-something-with mqueue))

Package [`posix-mqueue`], page 9,

Source [`lib.lisp`], page 6, (file)

5.1.3 Functions

`attributes` *QUEUE* [Function]
Retrieve attributes of the message queue.

Conditions:

BAD-FILE-DESCRIPTOR-INVALID

The message queue file descriptor (MQD) is invalid.

Package [`posix-mqueue`], page 9,

Source [`lib.lisp`], page 6, (file)

`buffer` *INSTANCE* [Function]
`(setf buffer)` *VALUE INSTANCE* [Function]

Package [`posix-mqueue`], page 9,

Source [`queue.lisp`], page 7, (file)

`close-queue` *QUEUE* [Function]
Close the message queue.

Conditions:

BAD-FILE-DESCRIPTOR-INVALID

The message queue file descriptor (MQD) is invalid.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`create-modesp` *THING* [Function]
Check if THING is a list and contains only MODEs.

Package [posix-mqueue], page 9,

Source [types.lisp], page 8, (file)

`current-messages` *INSTANCE* [Function]
`(setf current-messages)` *VALUE INSTANCE* [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`default-sizes` () [Function]
Return default sizes of a queue in a form (MAX-MESSAGES . MESSAGE-SIZE). This is done by creating a queue with a random name and by extracting its attributes. By using a 255 length name, we protect ourselves from name collision.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`max-messages` *INSTANCE* [Function]
`(setf max-messages)` *VALUE INSTANCE* [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`message-size` *INSTANCE* [Function]
`(setf message-size)` *VALUE INSTANCE* [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`non-blocking-p` *INSTANCE* [Function]
`(setf non-blocking-p)` *VALUE INSTANCE* [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`open-flagsp` *THING* [Function]
Check if THING is a list and contains only OFLAGs. Also, check that single-flags are present only once.

Package [posix-mqueue], page 9,

Source [types.lisp], page 8, (file)

`open-queue` *NAME* **&key** *OPEN-FLAGS CREATE-MODES MAX-MESSAGES MESSAGE-SIZE* [Function]

Create a new POSIX message queue or open an existing queue.

NAME is a string that identifies a queue. It MUST start with a slash (`"/"`) and MUST NOT contain other slashes. Example: `"/myqueue"`.

OPEN-FLAGS is a list of flags that control the operation of queue. Exactly one of the following must be specified in OPEN-FLAGS:

:read-only

Open the queue to receive messages only.

:write-only

Open the queue to send messages only.

:read-write

Open the queue to both send and receive messages.

Zero or more of the following flags:

:close-on-exec

Set the close-on-exec flag for the message queue descriptor. See open(2) for a discussion of why this flag is useful.

:create

Create the message queue if it does not exist. The owner (user ID) of the message queue is set to the effective user ID of the calling process. The group ownership (group ID) is set to the effective group ID of the calling process.

:exclusive

If :create was specified in OPEN-FLAGS, and a queue with the given name already exists, then fail signaling FILE-EXISTS condition.

:non-blocking

Open the queue in nonblocking mode. In circumstances where RECEIVE and SEND operations would normally block, these operations will return :try-again instead.

If :create is specified in OPEN-FLAGS, then three additional arguments can be supplied. The MODE argument specifies the permissions to be placed on the new queue. It is a list of

the following possible flags:

:user-read :user-write :group-read :group-write :other-read :other-read

In addition, MAX-MESSAGES and MESSAGE-SIZE specify the maximum number of messages and the maximum size of messages that the queue will allow. Usually, they default to their maximum values, 10 and 8192 respectively, but these values can be changes through /proc/sys/fs/mqueue/ interface. They must be provided in pair, as in the mq_open(3), but DEFAULT-SIZES function is provided to get default sizes of a queue.

This function can signal the following conditions:

ACCESS-DENIED-PERMISSION

The queue exists, but the caller does not have permission to open it in the specified mode.

ACCESS-DENIED-SLASHES

NAME contained more than one slash.

FILE-EXISTS

Both :create and :exclusive were specified in OPEN-FLAGS, but a queue with this NAME already exists.

INVALID-ARGUMENT-NAME

NAME doesn't follow the format described in mq_overview(7).

INVALID-ARGUMENT-SIZES

:create was specified in OPEN-FLAGS, but MAX-MESSAGES or MESSAGE-SIZE were invalid. Both of these fields must be greater than zero. In a process that is unprivileged (does not have the CAP_SYS_RESOURCE capability), MAX-MESSAGES must be less than or equal to the msg_max limit, and MESSAGE-SIZE must be less than or equal to the msgsize_max limit. In addition, even in a privileged process, MAX-MESSAGES cannot exceed the HARD_MAX limit. (See mq_overview(7) for details of these limits.).

Both of these limits can be changed through the /proc/sys/fs/mqueue/ interface.

TOO-MANY-OPEN-FILES

The per-process limit on the number of open file and message queue descriptors has been reached (see the description of RLIMIT_NOFILE in getrlimit(2)).

NAME-TOO-LONG

NAME was too long.

FILE-TABLE-OVERFLOW

The system-wide limit on the total number of open files and message queues has been reached.

NO-FILE-OR-DIRECTORY-JUST-SLASH

NAME was just "/" followed by no other characters.

NO-FILE-OR-DIRECTORY-NO-CREATE

The :create flag was not specified in OPEN-FLAGS, and no queue with this NAME exists.

OUT-OF-MEMORY

Insufficient memory.

NO-SPACE-LEFT-ON-DEVICE

Insufficient space for the creation of a new message queue. This probably occurred because the queues_max limit was encountered; see mq_overview(7).
SIMPLE-ERROR

This one can be signalled if the OPEN-FLAGS or the MODE are invalid.

BAD-FILE-DESCRIPTOR

The message queue file descriptor (MQD) is invalid. This is an internal error that should not happen, it is mainly for the writer of this library.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`receive` *QUEUE* [Function]

Remove the oldest message with the highest priority from the message QUEUE and return it as '(ARRAY (UNSIGNED-BYTE 8)). Return the priority associated with the received message as second value. Return the message LENGTH as a third value. Message length could be less than the returned BUFFER length. In fact, this is the same buffer used internally in queue to receive all messages. This function is provided for better control of the message data. Most library users would like to use RECEIVE-STRING, or RECEIVE-BUFFER, or RECEIVE-DISPLACED, instead.

If the queue is empty, then, by default, RECEIVE blocks until a message becomes available, or the call is interrupted by a signal handler. If the :non-blocking OPEN-FLAG is enabled for the message queue, then the call instead returns immediately with :try-again.

Conditions:

BAD-FILE-DESCRIPTOR-INVALID

The file descriptor specified MQD was invalid or not opened for reading.
INTERRUPTED-SYSTEM-CALL

The call was interrupted by a signal handler; see signal(7).

MESSAGE-TOO-LONG-ON-RECEIVE

Message length was less than the :message-size attribute of the message queue. This is an intarnal error that should not happen, it is mainly for the writer of this library.

Restarts:

RETRY-ON-INTERRUPT

If the call was interrupted by a signal handler, you can restart the call.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`receive-buffer` *QUEUE* [Function]
Behaves just luke RECEIVE, except that it creates a new buffer with ONLY message data.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`receive-displaced` *QUEUE* [Function]
Behaves just like RECEIVE, except that it tries to return a displaced array from internal buffer. You should not use it in a thread, unless protected by a lock.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`receive-string` *QUEUE* [Function]
Behaves just like RECEIVE, except that it tries to convert received message to string.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`send` *QUEUE MESSAGE-BUFFER PRIORITY* **&optional** *LENGTH* [Function]
Adds the MESSAGE-BUFFER to the message QUEUE. MESSAGE-BUFFER length must be less than or equal to the QUEUE's :message-size attribute. Zero-length messages are allowed. MESSAGE-BUFFER must be an '(array (unsigned-byte 8)). Additional LENGTH argument can be provided to limit the message being sent. By default, it is equal to MESSAGE-BUFFER length.

The PRIORITY argument is a nonnegative integer that specifies the priority of new message. Messages are placed on the QUEUE in decreasing order of priority, with newer messages of the same priority being placed after older messages with the same priority. See mq_overview(7) for details on the range for the message priority.

If the message QUEUE is already full (i.e., the number of messages on the QUEUE equals the QUEUE's :max-messages attribute), then, by default, SEND blocks until sufficient space becomes available to allow the message to be queued, or until the call is interrupted by a signal handler. If the :non-blocking flag is enabled for the message QUEUE, then the call instead returns :try-again.

Note: if you don't want to create a new buffer for sending to save space, you can reuse QUEUE's buffer. Use BUFFER function on a QUEUE to get it. Remember, that its data will be overwritten on next receive call.

Conditions:

BAD-FILE-DESCRIPTOR-ON-SEND

The file descriptor specified MQD was invalid or not opened for writing.
INTERRUPTED-SYSTEM-CALL

The call was interrupted by a signal handler; see signal(7).

MESSAGE-TOO-LONG-ON-SEND

MESSAGE length was greater than the :message-size attribute of the message QUEUE.

Restarts:

RETRY-ON-INTERRUPT

If the call was interrupted by a signal handler, you can restart the call.

Package [`posix-mqueue`], page 9,

Source [`lib.lisp`], page 6, (file)

`send-string` *QUEUE MESSAGE-STRING PRIORITY* [Function]
Behaves just like SEND, except that it sends a string, not an '(array (unsigned-byte 8))

Package [`posix-mqueue`], page 9,

Source [`lib.lisp`], page 6, (file)

`set-non-blocking` *QUEUE NON-BLOCKING-P* [Function]
Modify NON-BLOCKING-P attribute of the message queue.

Conditions:

BAD-FILE-DESCRIPTOR-INVALID

The message queue file descriptor (MQD) is invalid.

INVALID-ARGUMENT-ATTRIBUTES

mq-flags contained flags other than :non-blocking. This is an internal error that should not happen, it is mainly for the writer of this library.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`timed-receive` *QUEUE TIMESTAMP* [Function]

Behaves just like RECEIVE, except that if the queue is empty and the :non-blocking OPEN-FLAG is not enabled for the message queue, then the TIMESTAMP specifies how long the call will block. The TIMESTAMP is absolute, not relative. If no message is available, and the timeout has already expired by the time of the call, TIMED-RECEIVE returns immediately with :connection-timed-out.

Look LOCAL-TIME package for more information on timestamps.

Additional conditions:

INVALID-ARGUMENT-ON-SEND-RECEIVE

The call would have blocked, and timeout arguments were invalid, either because :sec was less than zero, or because :nsec was less than zero or greater than 1000 million.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`timed-receive-buffer` *QUEUE TIMESTAMP* [Function]

Behaves just like TIMED-RECEIVE, except that it creates a new buffer with ONLY message data.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`timed-receive-displaced` *QUEUE TIMESTAMP* [Function]

Behaves just like TIMED-RECEIVE, except that it tries to return a displaced array from internal buffer. You should not use it in a thread, unless protected by a lock.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`timed-receive-string` *QUEUE TIMESTAMP* [Function]

Behaves just like TIMED-RECEIVE, except that it tries to convert received message to string.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`timed-send` *QUEUE MESSAGE-BUFFER PRIORITY TIMESTAMP* [Function]
&optional *LENGTH*

Behaves just like SEND, except that if the QUEUE is full and the :non-blocking flag is not enabled for the message queue, then TIMESTAMP specifies how long the call will block. The TIMESTAMP is absolute, not relative. If the message queue is full, and the timeout has already expired by the time of the call, TIMED-SEND returns immediately with :connection-timed-out.

Look LOCAL-TIME package for more information on timestamps.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`timed-send-string` *QUEUE MESSAGE-STRING PRIORITY TIMESTAMP* [Function]

Behaves just like TIMED-SEND, except that it sends a string, not an '(array (unsigned-byte 8))

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

`unlink` *NAME* [Function]

Remove the specified message queue NAME. The message queue NAME is removed immediately. The queue itself is destroyed once any other processes that have the queue open close their descriptors referring to the queue.

Conditions:

ACCESS-DENIED-ON-UNLINK

The caller does not have permission to unlink this message queue.

NAME-TOO-LONG

NAME was too long.

NO-FILE-OR-DIRECTORY-ON-UNLINK

There is no message queue with the given NAME.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

5.1.4 Conditions

`access-denied-on-unlink` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses

[access-denied], page 32, (condition)

Direct Default Initargs

Initarg	Value
:message	"the caller does not have permission to unlink this message queue."

`access-denied-permission` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[access-denied], page 32, (condition)

Direct Default Initargs

Initarg	Value
:message	"the queue exists, but the caller does not have permission to open it in the specified mode."

`access-denied-slashes` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[access-denied], page 32, (condition)

Direct Default Initargs

Initarg	Value
:message	"name contained more than one slash."

`bad-file-descriptor-invalid` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[bad-file-descriptor], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"the message queue file descriptor (mqd) is invalid."

`bad-file-descriptor-on-receive` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[bad-file-descriptor], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"the file descriptor specified mqd was invalid or not opened for reading."

`bad-file-descriptor-on-send` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[bad-file-descriptor], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"the file descriptor specified mqd was invalid or not opened for writing."

`file-exists` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"both :create and :exclusive were specified in open-flags, but a queue with this name already exists."
:strerror	"file exists"

`file-table-overflow` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"the system-wide limit on the total number of open files and message queues has been reached."
:strerror	"too many open files in system"

`interrupted-system-call` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"the call was interrupted by a signal handler; see signal(7)."
:strerror	"interrupted system call"

invalid-argument-attributes () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[invalid-argument], page 34, (condition)

Direct Default Initargs

Initarg	Value
:message	"mq-flags contained flags other than :non-blocking."

invalid-argument-name () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[invalid-argument], page 34, (condition)

Direct Default Initargs

Initarg	Value
:message	"name doesn't follow the format described in mq_overview(7)."

invalid-argument-on-send-receive () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[invalid-argument], page 34, (condition)

Direct Default Initargs

Initarg	Value
:message	"the call would have blocked, and timeout arguments were invalid, either because :sec was less than zero, or because :nsec was less than zero or greater than 1000 million."

invalid-argument-on-unlink () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[invalid-argument], page 34, (condition)

Direct Default Initargs

Initarg	Value
`:message`	`"the caller does not have permission to unlink this message queue."`

`invalid-argument-sizes` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[invalid-argument], page 34, (condition)

Direct Default Initargs

Initarg	Value
`:message`	`":create was specified in open-flags, but max-messages or message-size was invalid. both of these fields must be greater than zero. in a process that is unprivileged (does not have the cap_sys_resource capability), max-messages must be less than or equal to the msg_max limit, and message-size must be less than or equal to the msgsize_max limit. in addition, even in a privileged process, :max-messages cannot exceed the hard_max limit. (see mq_overview(7) for details of these limits.). both of these limits can be changed through the /proc/sys/fs/mqueue/ interface."`

`message-too-long-on-receive` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[message-too-long], page 34, (condition)

Direct Default Initargs

Initarg	Value
`:message`	`"message length was less than the :message-size attribute of the message queue."`

`message-too-long-on-send` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[message-too-long], page 34, (condition)

Direct Default Initargs

Initarg	Value
`:message`	`"message length was greater than the :message-size attribute of the message queue."`

`name-too-long` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct Default Initargs

Initarg	Value
`:message`	`"name was too long."`
`:strerror`	`"file name too long"`

`no-file-or-directory-just-slash` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[no-file-or-directory], page 34, (condition)

Direct Default Initargs

Initarg	Value
`:message`	`"name was just \"/\" followed by no other characters."`

`no-file-or-directory-no-create` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[no-file-or-directory], page 34, (condition)

Direct Default Initargs

Initarg	Value
`:message`	`"the :create flag was not specified in open-flags, and no queue with this name exists."`

`no-file-or-directory-on-unlink` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[no-file-or-directory], page 34, (condition)

Direct Default Initargs

Initarg	Value
:message	"there is no message queue with the given name."

no-space-left-on-device () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"insufficient space for the creation of a new message queue. this probably occurred because the queues_max limit was encountered; see mq_overview(7)."
:strerror	"no space left on device"

out-of-memory () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"insufficient memory."
:strerror	"cannot allocate memory"

too-many-open-files () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct Default Initargs

Initarg	Value
:message	"the per-process limit on the number of open file and message queue descriptors has been reached (see the description of rlimit_nofile in getrlimit(2))."
:strerror	"too many open files"

5.1.5 Structures

`attributes` () [Structure]

POSIX message queue attributes.

Slot NON-BLOCKING-P indicates whether the receive/send operations would block. Slot MAX-MESSAGES shows queue's max number of messages. MESSAGE-SIZE is queue's message size. CURRENT-MESSAGES shows how much messages there are on queue now.

Package [`posix-mqueue`], page 9,

Source [`queue.lisp`], page 7, (file)

Direct superclasses

`structure-object` (structure)

Direct slots

`non-blocking-p` [Slot]

Type `boolean`

Readers [`non-blocking-p`], page 14, (function)

Writers [`(setf non-blocking-p)`], page 14, (function)

`max-messages` [Slot]

Type `(unsigned-byte 64)`

Initform `0`

Readers [`max-messages`], page 14, (function)

Writers [`(setf max-messages)`], page 14, (function)

`message-size` [Slot]

Type `(unsigned-byte 64)`

Initform `0`

Readers [`message-size`], page 14, (function)

Writers [`(setf message-size)`], page 14, (function)

`current-messages` [Slot]

Type `(unsigned-byte 64)`

Initform `0`

Readers [`current-messages`], page 14, (function)

Writers [`(setf current-messages)`], page 14, (function)

`queue` () [Structure]

Main type used to interact with POSIX message queues. It contains a queue's file descriptor (MQD) and a BUFFER used to receive messages.

It has a MQD slot: message queue's file descriptor. And a BUFFER slot: buffer used to receive messages form queue.

Package [`posix-mqueue`], page 9,

Source [`queue.lisp`], page 7, (file)

Direct superclasses

`structure-object` (structure)

Direct slots

`mqd` [Slot]

Type `(unsigned-byte 32)`

Initform `(error "mqueue descriptor was not provided")`

Readers [`mqd`], page 32, (function)

Writers [`(setf mqd)`], page 32, (function)

`buffer` [Slot]

Type `(array (unsigned-byte 8))`

Initform `(make-array 0 :element-type (quote (unsigned-byte 8)))`

Readers [`buffer`], page 13, (function)

Writers [`(setf buffer)`], page 13, (function)

5.1.6 Types

`create-modes` () [Type]

Type used to describe CREATE-MODES in OPEN-QUEUE.

Package [`posix-mqueue`], page 9,

Source [`types.lisp`], page 8, (file)

`open-flags` () [Type]

Type used to describe OPEN-FLAGS in OPEN-QUEUE.

Package [`posix-mqueue`], page 9,

Source [`types.lisp`], page 8, (file)

5.2 Internal definitions

5.2.1 Symbol macros

`*errno*` [Symbol Macro]

Package [`posix-mqueue`], page 9,

Source [`translation.lisp`], page 8, (file)

Expansion `(posix-mqueue::%var-accessor-*errno*)`

5.2.2 Macros

`%receive` *RECEIVE-FN CURRENT-FN RETURN-FORM* **&rest** *CURRENT-FN-ARGS* [Macro]

Macro used for generating various receive functions.

RECEIVE-FN is a function called to receive a message. CURRENT-FN is a function which will be called on interrupt. RETURN-FORM is a form placed at the end of the macro. It has access to BUFFER, LENGTH (of received message) and PRIORITY (of received message). CURRENT-FN-ARGS are additional arguments placed at the end of RECEIVE-FN and CURRENT-FN.

Package [`posix-mqueue`], page 9,

Source [lib.lisp], page 6, (file)

`%send` *SEND-FN CURRENT-FN* **&rest** *SEND-FN-ARGS* [Macro]
A macro used to generate send functions. SEND-FN is a function used to send the actual message. CURRENT-FN is a function which is called on interrupt. SEND-FN-ARGS are additional arguments placed at the end of SEND-FN and CURRENT-FN call.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

5.2.3 Functions

`%var-accessor-*errno*` () [Function]
`(setf %var-accessor-*errno*)` *VALUE* [Function]

Package [posix-mqueue], page 9,

Source [translation.lisp], page 8, (file)

`attributes-p` *OBJECT* [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`copy-attributes` *INSTANCE* [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`make-attributes` **&key** (*NON-BLOCKING-P* **NON-BLOCKING-P**) [Function]
(*MAX-MESSAGES* **MAX-MESSAGES**) (*MESSAGE-SIZE* **MESSAGE-SIZE**)
(*CURRENT-MESSAGES* **CURRENT-MESSAGES**)

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`make-queue` **&key** (*MQD* **MQD**) (*BUFFER* **BUFFER**) [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`mq-close` *MQDES* [Function]
Close POSIX message queue. See mq_close(3) for more details.

Package [posix-mqueue], page 9,

Source [spec.lisp], page 7, (file)

`mq-close-default` *MQDES* [Function]
Close default POSIX message queue. See mq_close(3).

Package [posix-mqueue], page 9,

Source [spec.lisp], page 7, (file)

`mq-getattr` *MQDES ATTR* [Function]
Get POSIX message queue attributes. See mq_getattr(3) for more details.

Package [posix-mqueue], page 9,

Source [spec.lisp], page 7, (file)

`mq-getattr-default` *MQDES ATTR* [Function]
Get POSIX message queue default attributes. See mq_getattr(3).

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mq-open` *NAME OFLAG MODE ATTR* [Function]
Open POSIX message queue. See mq_open(3) for more details.

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mq-open-default` *NAME OFLAG MODE ATTR* [Function]
Open POSIX message queue with default attributes. See mq_open(3).

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mq-receive` *MQDES MSG-PTR MSG-LEN MSG-PRIO* [Function]
Receive a message from POSIX message queue. See mq_receive(3) for more details.

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mq-send` *MQDES MSG-PTR MSG-LEN MSG-PRIO* [Function]
Send a message to POSIX message queue. See mq_send(3) for more details.

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mq-setattr` *MQDES NEWATTR OLDATTR* [Function]
Set POSIX message queue non-blocking attribute. See mq_setattr(3) for more details.

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mq-timedreceive` *MQDES MSG-PTR MSG-LEN MSG-PRIO ABS-TIMEOUT* [Function]
Receive a message from POSIX message queue. See mq_timedreceive(3) for more details.

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mq-timedsend` *MQDES MSG-PTR MSG-LEN MSG-PRIO ABS-TIMEOUT* [Function]
Send a message to POSIX message queue. See mq_timedsend(3) for more details.

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mq-unlink` *NAME* [Function]
Unlink POSIX message queue. See mq_unlink(3) for more details.

Package [`posix-mqueue`], page 9,

Source [`spec.lisp`], page 7, (file)

`mqd` *INSTANCE* [Function]
`(setf mqd)` *VALUE INSTANCE* [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`queue-p` *OBJECT* [Function]

Package [posix-mqueue], page 9,

Source [queue.lisp], page 7, (file)

`random-queue-name` **&key** *LENGTH START END* [Function]
Generate random queue name with specified LENGTH, with characters starting from START to END. With slash at the beginning.

Package [posix-mqueue], page 9,

Source [lib.lisp], page 6, (file)

5.2.4 Generic functions

`message` *CONDITION* [Generic Function]

Package [posix-mqueue], page 9,

Methods

`message` (*CONDITION* `generic`) [Method]

Source [condition.lisp], page 5, (file)

`strerror` *CONDITION* [Generic Function]

Package [posix-mqueue], page 9,

Methods

`strerror` (*CONDITION* `generic`) [Method]

Source [condition.lisp], page 5, (file)

5.2.5 Conditions

`access-denied` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct subclasses

- [access-denied-permission], page 22, (condition)
- [access-denied-slashes], page 22, (condition)
- [access-denied-on-unlink], page 21, (condition)

Direct Default Initargs

Initarg	Value
`:strerror`	`"permission denied"`

`bad-file-descriptor` () [Condition]

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
[generic], page 33, (condition)

Direct subclasses
- [bad-file-descriptor-invalid], page 22, (condition)
- [bad-file-descriptor-on-receive], page 22, (condition)
- [bad-file-descriptor-on-send], page 23, (condition)

Direct Default Initargs

Initarg	Value
:strerror	"bad file descriptor"

`generic` () [Condition]

Generic error used as the base for all conditions. Must contain STRERROR and MESSAGE.

Package [posix-mqueue], page 9,

Source [condition.lisp], page 5, (file)

Direct superclasses
`error` (condition)

Direct subclasses
- [out-of-memory], page 27, (condition)
- [file-exists], page 23, (condition)
- [file-table-overflow], page 23, (condition)
- [too-many-open-files], page 27, (condition)
- [no-space-left-on-device], page 27, (condition)
- [name-too-long], page 26, (condition)
- [interrupted-system-call], page 23, (condition)
- [no-file-or-directory], page 34, (condition)
- [bad-file-descriptor], page 33, (condition)
- [access-denied], page 32, (condition)
- [invalid-argument], page 34, (condition)
- [message-too-long], page 34, (condition)

Direct methods
- [message], page 32, (method)
- [strerror], page 32, (method)

Direct slots

`strerror` [Slot]

Error string from CFFI's strerror.

Initargs :strerror

Readers [strerror], page 32, (generic function)

`message` [Slot]

More specific message string.

Initargs `:message`

Readers [`message`], page 32, (generic function)

`invalid-argument` () [Condition]

Package [`posix-mqueue`], page 9,

Source [`condition.lisp`], page 5, (file)

Direct superclasses

[`generic`], page 33, (condition)

Direct subclasses

- [`invalid-argument-name`], page 24, (condition)
- [`invalid-argument-sizes`], page 25, (condition)
- [`invalid-argument-attributes`], page 24, (condition)
- [`invalid-argument-on-unlink`], page 24, (condition)
- [`invalid-argument-on-send-receive`], page 24, (condition)

Direct Default Initargs

Initarg	Value
`:strerror`	`"invalid argument"`

`message-too-long` () [Condition]

Package [`posix-mqueue`], page 9,

Source [`condition.lisp`], page 5, (file)

Direct superclasses

[`generic`], page 33, (condition)

Direct subclasses

- [`message-too-long-on-receive`], page 25, (condition)
- [`message-too-long-on-send`], page 25, (condition)

Direct Default Initargs

Initarg	Value
`:strerror`	`"message too long"`

`no-file-or-directory` () [Condition]

Package [`posix-mqueue`], page 9,

Source [`condition.lisp`], page 5, (file)

Direct superclasses

[`generic`], page 33, (condition)

Direct subclasses

- [`no-file-or-directory-just-slash`], page 26, (condition)
- [`no-file-or-directory-no-create`], page 26, (condition)
- [`no-file-or-directory-on-unlink`], page 26, (condition)

Direct Default Initargs

Initarg	Value
`:strerror`	`"no such file or directory"`

5.2.6 Classes

`mq-attr-tclass` () [Class]

Package [posix-mqueue], page 9,

Source [types.lisp], page 8, (file)

Direct superclasses

- `translatable-foreign-type` (class)
- `foreign-struct-type` (class)

`mq-get-attr-type` () [Class]

Type used to pass ATTRIBUTES as C-function argument. Translation maps ATTRIBUTES to MQ-ATTR CStruct.

Package [posix-mqueue], page 9,

Source [types.lisp], page 8, (file)

Direct superclasses

`enhanced-foreign-type` (class)

Direct methods

`expand-to-foreign-dyn` (method)

Direct Default Initargs

Initarg	Value
`:actual-type`	`(quote (:pointer))`

`mq-non-blocking-attr-type` () [Class]

Type used to get attributes through a pointer. To fill a CStruct through a pointer passed to function. Translation for this type does exactly this, at the end of the function call, it fills Lisp class with values from MQ-ATTR.

Package [posix-mqueue], page 9,

Source [types.lisp], page 8, (file)

Direct superclasses

`enhanced-foreign-type` (class)

Direct methods

`expand-to-foreign-dyn` (method)

Direct Default Initargs

Initarg	Value
`:actual-type`	`(quote (:pointer))`

`mq-size-attr-type` () [Class]

Type used to pass ATTRIBUTES as C-function argument. Translation maps ATTRIBUTES to MQ-ATTR CStruct.

Package [posix-mqueue], page 9,

Source [types.lisp], page 8, (file)

Direct superclasses

`enhanced-foreign-type` (class)

Direct methods
`expand-to-foreign-dyn` (method)

Direct Default Initargs

Initarg	Value
`:actual-type`	`(quote (:pointer))`

`mqd-type` () [Class]
Type used to describe POSIX message queue file descriptor. Also, there are translations defined for this type (:int) from QUEUE class.

Package [`posix-mqueue`], page 9,

Source [`types.lisp`], page 8, (file)

Direct superclasses
`enhanced-foreign-type` (class)

Direct methods
`expand-to-foreign` (method)

Direct Default Initargs

Initarg	Value
`:actual-type`	`(quote (:int))`

`result-type` () [Class]
Type used to describe C-style result of functions. There is a translation that maps -1 to keyword representation of the error through the errno.

Package [`posix-mqueue`], page 9,

Source [`types.lisp`], page 8, (file)

Direct superclasses
`enhanced-foreign-type` (class)

Direct methods
`expand-from-foreign` (method)

Direct Default Initargs

Initarg	Value
`:actual-type`	`(quote (:int))`

`timespec-tclass` () [Class]

Package [`posix-mqueue`], page 9,

Source [`types.lisp`], page 8, (file)

Direct superclasses
- `translatable-foreign-type` (class)
- `foreign-struct-type` (class)

`timespec-type` () [Class]
Type used to pass LOCAL-TIME:TIMESTAMP as C timespec.

Package [`posix-mqueue`], page 9,

Source [`types.lisp`], page 8, (file)

Direct superclasses
`enhanced-foreign-type` (class)

Direct methods
`expand-to-foreign-dyn` (method)

Direct Default Initargs

Initarg	Value
`:actual-type`	`(quote (:pointer))`

Appendix A Indexes

A.1 Concepts

C

F

L

M

A.2 Functions

%

(

A

B

C

D

F

G

M

N

O

Q

R

S

T

U

W

A.3 Variables

*

B

C

M

N

S

A.4 Data types

A

B

C

F

G

I

M

N

O

P

Q

R

S

T