



*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 'mq\_unlink.3p' command***

***\$ man mq\_unlink.3p***

MQ\_UNLINK(3P)      POSIX Programmer's Manual      MQ\_UNLINK(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

mq\_unlink ? remove a message queue (REALTIME)

### SYNOPSIS

```
#include <mqueue.h>

int mq_unlink(const char *name);
```

### DESCRIPTION

The mq\_unlink() function shall remove the message queue named by the string name. If one or more processes have the message queue open when mq\_unlink() is called, destruction of the message queue shall be postponed until all references to the message queue have been closed. However, the mq\_unlink() call need not block until all references have been closed; it may return immediately.

After a successful call to mq\_unlink(), reuse of the name shall subsequently cause mq\_open() to behave as if no message queue of this name exists (that is, mq\_open() will fail if O\_CREAT is not set, or will create a new message queue if O\_CREAT is set).

### RETURN VALUE

Upon successful completion, the function shall return a value of zero.

Otherwise, the named message queue shall be unchanged by this function call, and the function shall return a value of -1 and set `errno` to indicate the error.

## ERRORS

The `mq_unlink()` function shall fail if:

`EACCES` Permission is denied to unlink the named message queue.

`EINTR` The call to `mq_unlink()` blocked waiting for all references to the named message queue to be closed and a signal interrupted the call.

`ENOENT` The named message queue does not exist.

The `mq_unlink()` function may fail if:

### `ENAMETOOLONG`

The length of the name argument exceeds `{_POSIX_PATH_MAX}` on systems that do not support the XSI option or exceeds `{_XOPEN_PATH_MAX}` on XSI systems, or has a pathname component that is longer than `{_POSIX_NAME_MAX}` on systems that do not support the XSI option or longer than `{_XOPEN_NAME_MAX}` on XSI systems. A call to `mq_unlink()` with a name argument that contains the same message queue name as was previously used in a successful `mq_open()` call shall not give an `[ENAMETOOLONG]` error.

The following sections are informative.

## EXAMPLES

None.

## APPLICATION USAGE

None.

## RATIONALE

None.

## FUTURE DIRECTIONS

A future version might require the `mq_open()` and `mq_unlink()` functions to have semantics similar to normal file system operations.

## SEE ALSO

mq\_close(), mq\_open(), msgctl(), msgget(), msgrcv(), msgsnd()

The Base Definitions volume of POSIX.1-2017, <mqqueue.h>

## COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html).

IEEE/The Open Group

2017

MQ\_UNLINK(3P)