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 'msgget.3p' command

$ man msgget.3p

MSGGET(3P) POSIX Programmer's Manual MSGGET(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

msgget ? get the XSI message queue identifier

SYNOPSIS

#include <sys/msg.h>

int msgget(key_t key, int msgflg);

DESCRIPTION

The msgget() function operates on XSI message queues (see the Base Def? initions volume of POSIX.1?2017, Section 3.226, Message Queue). It is unspecified whether this function interoperates with the realtime in? terprocess communication facilities defined in Section 2.8, Realtime.

The msgget() function shall return the message queue identifier associ? ated with the argument key.

A message queue identifier, associated message queue, and data struc? ture (see <sys/msg.h>), shall be created for the argument key if one of the following is true:

* The argument key is equal to IPC_PRIVATE.
* The argument key does not already have a message queue identifier

associated with it, and (msgflg & IPC_CREAT) is non-zero.

Upon creation, the data structure associated with the new message queue identifier shall be initialized as follows:

* msg_perm.cuid, msg_perm.uid, msg_perm.cgid, and msg_perm.gid shall be set to the effective user ID and effective group ID, respectively, of the calling process.
* The low-order 9 bits of msg_perm.mode shall be set to the low-order 9 bits of msgflg.
* msg_qnum, msg_lspid, msg_lrpid, msg_stime, and msg_rtime shall be set to 0.
* msg_ctime shall be set to the current time, as described in Section 2.7.1, IPC General Description.
* msg_qbytes shall be set to the system limit.

RETURN VALUE

Upon successful completion, msgget() shall return a non-negative integer, namely a message queue identifier. Otherwise, it shall return -1 and set errno to indicate the error.

ERRORS

The msgget() function shall fail if:

EACCES A message queue identifier exists for the argument key, but operation permission as specified by the low-order 9 bits of msgflg would not be granted; see Section 2.7, XSI Interprocess Communication.

EEXIST A message queue identifier exists for the argument key but ((msgflg & IPC_CREAT) && (msgflg & IPC_EXCL)) is non-zero.

ENOENT A message queue identifier does not exist for the argument key and (msgflg & IPC_CREAT) is 0.

ENOSPC A message queue identifier is to be created but the system-imposed limit on the maximum number of allowed message queue identifiers system-wide would be exceeded.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The POSIX Realtime Extension defines alternative interfaces for inter? process communication (IPC). Application developers who need to use IPC should design their applications so that modules using the IPC routines described in Section 2.7, XSI Interprocess Communication can be easily modified to use the alternative interfaces.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.7, XSI Interprocess Communication, Section 2.8, Realtime, ftok(), mq_close(), mq_getattr(), mq_notify(), mq_open(), mq_receive(), mq_send(), mq_setattr(), mq_unlink(), msgctl(), msgrcv(), msgsnd()

The Base Definitions volume of POSIX.1?2017, Section 3.226, Message Queue, <sys_msg.h>