

May 12, 03 12:29	stdin	Page 1/2
MSGGET(2)	Linux Programmer's Manual	MSGGET(2)
NAME		
msgget - get a message queue identifier		
SYNOPSIS		
#include <sys/types.h> #include <sys/ipc.h> #include <sys/msg.h>		
int msgget(key_t key, int msgflg);		
DESCRIPTION		
<p>The function returns the message queue identifier associated to the value of the <i>key</i> argument. A new message queue is created if <i>key</i> has value IPC_PRIVATE or <i>key</i> isn't IPC_PRIVATE, no existing message queue is associated to <i>key</i>, and IPC_CREAT is asserted in <i>msgflg</i> (i.e. <i>msgflg</i> & IPC_CREAT is nonzero). The presence in <i>msgflg</i> of the fields IPC_CREAT and IPC_EXCL plays the same role, with respect to the existence of the message queue, as the presence of O_CREAT and O_EXCL in the mode argument of the open(2) system call: i.e. the msgget function fails if <i>msgflg</i> asserts both IPC_CREAT and IPC_EXCL and a message queue already exists for <i>key</i>.</p> <p>Upon creation, the lower 9 bits of the argument <i>msgflg</i> define the access permissions of the message queue. These permission bits have the same format and semantics as the access permissions parameter in open(2) or creat(2) system calls. (The execute permissions are not used.)</p> <p>Furthermore, while creating, the system call initializes the system message queue data structure msqid_ds as follows:</p> <p>msg_perm.cuid and msg_perm.uid are set to the effective user-ID of the calling process.</p> <p>msg_perm.cgid and msg_perm.gid are set to the effective group-ID of the calling process.</p> <p>The lowest order 9 bits of msg_perm.mode are set to the lowest order 9 bit of <i>msgflg</i>.</p> <p>msg_qnum, msg_lspid, msg_lrpid, msg_stime and msg_rtime are set to 0.</p> <p>msg_ctime is set to the current time.</p> <p>msg_qbytes is set to the system limit MSGMNB.</p> <p>If the message queue already exists the access permissions are verified, and a check is made to see if it is marked for destruction.</p>		
RETURN VALUE		
If successful, the return value will be the message queue identifier (a nonnegative integer), otherwise -1 with errno indicating the error.		
ERRORS		
For a failing return, errno will be set to one among the following values:		
EACCES	A message queue exists for <i>key</i> , but the calling process has no access permissions to the queue.	

May 12, 03 12:29	stdin	Page 2/2
EEXIST	A message queue exists for <i>key</i> and <i>msgflg</i> was asserting both IPC_CREAT and IPC_EXCL .	
EIDRM	The message queue is marked for removal.	
ENOENT	No message queue exists for <i>key</i> and <i>msgflg</i> wasn't asserting IPC_CREAT .	
ENOMEM	A message queue has to be created but the system has not enough memory for the new data structure.	
ENOSPC	A message queue has to be created but the system limit for the maximum number of message queues (MSGMNI) would be exceeded.	
NOTES		
IPC_PRIVATE isn't a flag field but a key_t type. If this special value is used for <i>key</i> , the system call ignores everything but the lowest order 9 bits of <i>msgflg</i> and creates a new message queue (on success).		
The following is a system limit on message queue resources affecting a msgget call:		
MSGMNI	System wide maximum number of message queues: policy dependent.	
BUGS		
Use of IPC_PRIVATE does not actually prohibit other processes from getting access to the allocated message queue.		
There is currently no intrinsic way for a process to ensure exclusive access to a message queue. Asserting both IPC_CREAT and IPC_EXCL in <i>msgflg</i> only ensures (on success) that a new message queue will be created, it doesn't imply exclusive access to the message queue.		
CONFORMING TO		
SVr4, SVID. SVr4 does not document the EIDRM error code.		
SEE ALSO		
ftok(3) , ipc(5) , msgctl(2) , msgsnd(2) , msgrcv(2)		
Linux 0.99.13	1993-11-01	MSGGET(2)