NAME

flushq – remove messages from a queue

SYNOPSIS

#include <sys/stream.h>

void flushq(queue_t $*q$**, int** *flag***);**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

ARGUMENTS

q Pointer to the queue to be flushed.

flag Valid *flag* values are:

- **FLUSHDATA** Flush only data messages (types **M_DATA M_DELAY M_PROTO** and **M_PCPROTO**).
- **FLUSHALL** Flush all messages.

DESCRIPTION

flushq( ) frees messages and their associated data structures by calling **freemsg**(9F). If the queue's count falls below the low water mark and the queue was blocking an upstream service procedure, the nearest upstream service procedure is enabled.

CONTEXT

flushq( ) can be called from user or interrupt context.

EXAMPLES

This example depicts the canonical flushing code for STREAMS modules. The module has a write service procedure and potentially has messages on the queue. If it receives an **M_FLUSH** message, and if the **FLUSHR** bit is on in the first byte of the message (line 10), then the read queue is flushed (line 11). If the **FLUSHW** bit is on (line 12), then the write queue is flushed (line 13). Then the message is passed along to the next entity in the stream (line 14). See the example for **qreply**(9F) for the canonical flushing code for drivers.

```
1  /*
2   * Module write-side put procedure.
3   */
4  xxxwput(q, mp)
5     queue_t *q;
6     mblk_t *mp;
7  {
8         switch(mp->b_datap->db_type) {
9           case M_FLUSH:
```

```
                if (*mp->b_rptr & FLUSHR)
                        flushq(RD(q), FLUSHALL);
                if (*mp->b_rptr & FLUSHW)
                        flushq(q, FLUSHALL);
                putnext(q, mp);
                break;
                . . .
        }
}
```

SEE ALSO

flushband(9F), **freemsg**(9F), **putq**(9F), **qreply**(9F)

Writing Device Drivers
STREAMS Programming Guide