NAME

mri_stat - Statistics collection entry point for rings

SYNOPSIS

#include <sys/mac_provider.h>

int
prefix_ring_stat(*mac_ring_driver_t rh, uint_t stat, uint64_t *val*);

INTERFACE LEVEL

Evolving - This interface is still evolving. API and ABI stability is not guaranteed.

PARAMETERS

rh A pointer to the ring's private data that was passed in via the *mri_driver* member of the mac_ring_info(9S) structure as part of the mr_rget(9E) entry point.

stat The numeric identifier of a statistic.

val A pointer to a 64-bit unsigned value in which the device driver should place statistic.

DESCRIPTION

The **mri_stat**() entry point is called by the MAC framework to get statistics that have been scoped to the ring, indicated by *rh*.

The set of statistics that the driver should check depends on the kind of ring that is in use. If the driver encounters an unknown statistic it should return ENOTSUP. All the statistics should be values that are scoped to the ring itself. This is in contrast to the normal mc_getstat(9E) entry point, which has statistics for the entire device. Other than the scoping, the statistics listed below have the same meaning as they do in the *STATISTICS* section of mac(9E). See mac(9E) for more detailed meanings of those statistics.

Receive rings should support the following statistics:

- MAC_STAT_IPACKETS
- MAC_STAT_RBYTES

Transmit rings should support the following statitics:

- MAC_STAT_OBYTES

- MAC_STAT_OPACKETS

EXAMPLES

The following example shows how a driver might structure its **mri_stat**() entry point.

```
#include <sys/mac_provider.h>

/*
 * Note, this example merely shows the structure of the function. For
 * the purpose of this example, we assume that we have a per-ring
 * structure which has members that indicate its stats and that it has a
 * lock which is used to serialize access to this data.
 */

static int
example_tx_ring_stat(mac_ring_driver_t rh, uint_t stat, uint64_t *val)
{
        example_tx_ring_t *etrp = arg;

        mutex_enter(&etrp->etrp_lock);
        switch (stat) {
        case MAC_STAT_OBYTES:
                *val = etrp->etrp_stats.eps_obytes;
                break;
        case MAC_STAT_OPACKETS:
                *val = etrp->etrp_stats.eps_opackets;
                break;
        default:
                mutex_exit(&etrp->etrp_lock);
                return (ENOTSUP);
        }
        mutex_exit(&etrp->etrp_lock);

        return (0);
}

static int
example_rx_ring_stat(mac_ring_driver_t rh, uint_t stat, uint64_t *val)
{
        example_rx_ring_t *errp = arg;
```

```
        mutex_enter(&errp->errp_lock);
        switch (stat) {
        case MAC_STAT_RBYTES:
                *val = errp->errp_stats.eps_ibytes;
                break;
        case MAC_STAT_IPACKETS:
                *val = errp->errp_stats.eps_ipackets;
                break;
        default:
                mutex_exit(&errp->errp_lock);
                return (ENOTSUP);
        }
        mutex_exit(&errp->errp_lock);

        return (0);
}
```

ERRORS

The device driver may return one of the following errors. While this list is not intended to be exhaustive, it is recommend to use one of these if possible.

ENOTSUP	The specified statistic is unknown, unsupported, or unimplemented.
EIO	A transport or DMA FM related error occurred while trying to sync data from the device.
ECANCELLED	The device is not currently in a state where it can currently service the request.

SEE ALSO

mac(9E), mac_capab_rings(9E), mc_getstat(9E), mr_rget(9E), mac_ring_info(9S)