NAME

mac_filter, **mgi_addmac**, **mgi_remmac**, **mgi_addvlan**, **mgi_remvlan**, **mgi_addmvf**, **mgi_remmvf** - add and remove filters from MAC groups

SYNOPSIS

#include <sys/mac_provider.h>

int
prefix_ring_add_mac(*mac_group_driver_t driver, const uint8_t *mac, uint_t flags*);

int
prefix_ring_rem_mac(*mac_group_driver_t driver, const uint8_t *mac, uint_t flags*);

int
prefix_ring_add_vlan(*mac_group_driver_t driver, uint16_t vlan, uint_t flags*);

int
prefix_ring_rem_vlan(*mac_group_driver_t driver, uint16_t vlan, uint_t flags*);

int
prefix_ring_add_macvlan(*mac_group_driver_t driver, const uint8_t *mac, uint16_t vlan, uint_t flags*);

int
prefix_ring_rem_macvlan(*mac_group_driver_t driver, const uint8_t *mac, uint16_t vlan, uint_t flags*);

INTERFACE LEVEL

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

PARAMETERS

driver A pointer to the ring's private data that was passed in via the *mgi_driver* member of the mac_group_info(9S) structure as part of the mr_gget(9E) entry point.

mac A pointer to an array of bytes that contains the unicast address to add or remove from a filter. It is guaranteed to be at least as long, in bytes, as the MAC plugin's address length. For Ethernet devices that length is six bytes, ETHERADDRL.

vlan The numeric value of the VLAN that should be added or removed from a filter.

flags This member is reserved for future use. Drivers should ensure that it is **0**.

DESCRIPTION

The **mac_filter**() family of entry points are used to add and remove various classes of filters from a device. For more information on the filters, see the *Filters* section of mac_capab_rings(9S).

The *driver* argument indicates which group the request to add or remove a filter is being requested on. The *flags* member contains values that modify the behavior of the filters. Currently this is reserved for future use. The driver must check the *flags* member. If any unknown or unsupported members values are present, then the driver should return ENOTSUP.

The filter addition operations, **mgi_addmac**(), **mgi_addvlan**(), and **mgi_addmvf**(), all instruct the system to add a filter to the specified group. The filter should not target any specific ring in the group. If multiple rings are present, then the driver should ensure that the hardware balances incoming traffic amongst all of the rings through a consistent hashing mechanism such as receive side scaling.

The **mgi_addmac**() entry point instructs the driver to add the MAC address specified in *mac* to the filter list for the group. The MAC address should always be a unicast MAC address; however, the driver is encouraged to verify that before adding it.

The **mgi_remmac**() should remove the MAC address specified in *mac* from the filter list for the group.

The **mgi_addvlan**() entry point instructs the driver to add the VLAN specified in *vlan* to the filter list for the group. The **mgi_remvlan**() entry point instructs the driver to remove the VLAN specified in *vlan* from the filter list for the group.

The **mgi_addmvl**() entry point instructs the driver to add a MAC address and VLAN tuple to the filter list for the group. This entry point should ensure that hardware only ever inserts values if it can match both the MAC address in *mac* and the VLAN in *vlan*. Importantly, it should only allow packets that match both the *mac* and *vlan*.

Stacking Filters

Multiple filters of the same class should always be treated as a logical-OR. The frame may match any of the filters in a given class to be accepted. Filters of different classes should always be treated as a logical-AND. The frame must match a filter in all programmed classes to be accepted. For more information, see the *Filters* section of mac_capab_rings(9S).

RETURN VALUES

Upon successful completion, the driver should ensure that the filter has been added or removed and return **0**. Otherwise, it should return the appropriate error number.

ERRORS

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

EEXIST	The requested filter has already been programmed into the device.
EINVAL	The address *mac* is not a valid unicast address. The VLAN *vlan* is not a valid VLAN identifier.
EIO	The driver encountered a device or transport error while trying to update the device's state.
ENOENT	The driver was asked to remove a filter which was not currently programmed.
ENOTSUP	An unknown flag is present in the **flags**() argument.
ENOSPC	The driver has run out of available hardware filters.

SEE ALSO

mac(9E), mac_capab_rings(9E), mr_gget(9E), mac_group_info(9S)