NAME

mc_unicst - set device unicast address

SYNOPSIS

#include <sys/mac_provider.h>

int
prefix_m_unicst(*void *driver, const uint8_t *mac*);

INTERFACE LEVEL

illumos DDI specific

PARAMETERS

driver A pointer to the driver's private data that was passed in via the **m_pdata** member of the mac_register(9S) structure to the mac_register(9F) function.

mac A pointer to an array of bytes that contains the new unicast address of the device. It is guaranteed to be at least a number of bytes long equal to the length of the MAC plugin's address length. For Ethernet devices that length is six bytes, **ETHERADDRL**.

DESCRIPTION

The **mc_unicst**() entry point is used by the MAC framework to indicate that the device driver should update the primary MAC address of the device. In the basic mode of operation, this entry point is required and the device has a single primary MAC address. If multiple MAC addresses are required, the device will be placed into promiscuous mode. This call should overwrite the existing MAC address that is programmed into the device.

As noted in the *PARAMETERS* section, the *mac* array is guaranteed to be at least as many bytes as is required to specify an address; however, it should be assumed to be no longer than that value.

The device driver can optionally assert that the address is in the valid form for a unicast address and then program the device. The device driver can access its device soft state by casting the *device* pointer to the appropriate structure. As this may be called while other operations are ongoing, the device driver should employ the appropriate locking while updating the data.

It is recommended that device drivers always maintain a copy of the current unicast address in its soft state so that way it can recover from various device reset and errors or handle requests to suspend and resume the device that may result in device registers being cleared.

Some devices support multiple MAC address filters. The **mc_unicst**() entry point only supports a single

MAC address. In this case, devices should only use a single MAC address and replace that MAC address. To enable the operating system to take advantage of multiple unicast MAC address filters, the driver should implement the MAC_CAPAB_RINGS capability. See mac_capab_rings(9E) for more information.

RETURN VALUES

Upon successful completion, the device driver should have updated its unicast filter and return **0**. Otherwise, the MAC address should remain unchanged and the driver should return an 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.

EINVAL The address *mac* is not a valid unicast address.

EIO The driver encountered a device or transport error while trying to update the device’s state.

SEE ALSO

mac(9E), mac_capab_rings(9E), mac_register(9F), mac_register(9S)