

NAME

em - Intel(R) PRO/1000 Gigabit Ethernet adapter driver

SYNOPSIS

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device iflib
device em
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
if_em_load="YES"
```

DESCRIPTION

The **em** driver provides support for PCI/PCI-X Gigabit Ethernet adapters based on the Intel 82540, 82541ER, 82541PI, 82542, 82543, 82544, 82545, 82546, 82546EB, 82546GB, and 82547 controller chips.

The **em** driver provides support for PCI Express Gigabit Ethernet adapters based on the Intel 82571, 82572, 82573, 82574, and 82583 Ethernet controller chips.

The **em** driver provides support for Gigabit Ethernet adapters connected to I/O Controller Hub (ICH) and Platform Controller Hub (PCH) including Intel 80003ES2LAN, 82562, 82566, 82567, 82577, 82578, 82579, i217, i218, and i219.

The **em** driver provides support for PCI Express Gigabit Ethernet adapters based on the Intel 82575, 82576, 82580, i210, i211, and i35x. These appear as **igb** interfaces to maintain compatibility with existing infrastructure.

The driver supports Transmit/Receive checksum offload and Jumbo Frames on all but 82542-based adapters.

Furthermore it supports TCP segmentation offload (TSO) on all adapters but those based on the 82542, 82543, 82544 and 82547 controller chips. The identification LEDs of the adapters supported by the **em** driver can be controlled via the led(4) API for localization purposes. For further hardware information, see the *README* included with the driver.

For questions related to hardware requirements, refer to the documentation supplied with your Intel PRO/1000 adapter. All hardware requirements listed apply to use with FreeBSD.

Support for Jumbo Frames is provided via the interface MTU setting. Selecting an MTU larger than 1500 bytes with the `ifconfig(8)` utility configures the adapter to receive and transmit Jumbo Frames. The maximum MTU size for Jumbo Frames is 16114.

This driver supports hardware assisted VLANs. The **em** driver supports the following media types:

autoselect Enables auto-negotiation for speed and duplex.

10baseT/UTP Sets 10Mbps operation. Use the **mediaopt** option to select **full-duplex** mode.

100baseTX Sets 100Mbps operation. Use the **mediaopt** option to select **full-duplex** mode.

1000baseSX Sets 1000Mbps operation. Only **full-duplex** mode is supported at this speed.

1000baseTX Sets 1000Mbps operation. Only **full-duplex** mode is supported at this speed.

The **em** driver supports the following media options:

full-duplex Forces full-duplex operation

half-duplex
Forces half-duplex operation.

Only use **mediaopt** to set the driver to **full-duplex**. If **mediaopt** is not specified, the driver defaults to **half-duplex**.

For more information on configuring this device, see `ifconfig(8)`.

HARDWARE

The **em** driver supports Gigabit Ethernet adapters based on the Intel 82540, 82541ER, 82541PI, 82542, 82543, 82544, 82545, 82546, 82546EB, 82546GB, 82547, 82571, 82572, 82573, 82574, 82575, 82576, and 82580 controller chips:

- ⌚ Intel Gigabit ET Dual Port Server Adapter (82576)
- ⌚ Intel Gigabit VT Quad Port Server Adapter (82575)
- ⌚ Intel Single, Dual and Quad Gigabit Ethernet Controller (82580)
- ⌚ Intel i210 and i211 Gigabit Ethernet Controller
- ⌚ Intel i350 and i354 Gigabit Ethernet Controller
- ⌚ Intel PRO/1000 CT Network Connection (82547)
- ⌚ Intel PRO/1000 F Server Adapter (82543)

- ⌘ Intel PRO/1000 Gigabit Server Adapter (82542)
- ⌘ Intel PRO/1000 GT Desktop Adapter (82541PI)
- ⌘ Intel PRO/1000 MF Dual Port Server Adapter (82546)
- ⌘ Intel PRO/1000 MF Server Adapter (82545)
- ⌘ Intel PRO/1000 MF Server Adapter (LX) (82545)
- ⌘ Intel PRO/1000 MT Desktop Adapter (82540)
- ⌘ Intel PRO/1000 MT Desktop Adapter (82541)
- ⌘ Intel PRO/1000 MT Dual Port Server Adapter (82546)
- ⌘ Intel PRO/1000 MT Quad Port Server Adapter (82546EB)
- ⌘ Intel PRO/1000 MT Server Adapter (82545)
- ⌘ Intel PRO/1000 PF Dual Port Server Adapter (82571)
- ⌘ Intel PRO/1000 PF Quad Port Server Adapter (82571)
- ⌘ Intel PRO/1000 PF Server Adapter (82572)
- ⌘ Intel PRO/1000 PT Desktop Adapter (82572)
- ⌘ Intel PRO/1000 PT Dual Port Server Adapter (82571)
- ⌘ Intel PRO/1000 PT Quad Port Server Adapter (82571)
- ⌘ Intel PRO/1000 PT Server Adapter (82572)
- ⌘ Intel PRO/1000 T Desktop Adapter (82544)
- ⌘ Intel PRO/1000 T Server Adapter (82543)
- ⌘ Intel PRO/1000 XF Server Adapter (82544)
- ⌘ Intel PRO/1000 XT Server Adapter (82544)

LOADER TUNABLES

Tunables can be set at the loader(8) prompt before booting the kernel or stored in loader.conf(5). See iflib(4) for per-instance variables.

hw.em.disable_crc_stripping

Disable or enable hardware stripping of CRC field. This is mostly useful on BMC/IPMI shared interfaces where stripping the CRC causes remote access over IPMI to fail. Default 0 (enabled).

hw.em.eee_setting

Disable or enable Energy Efficient Ethernet. Default 1 (disabled).

hw.em.smart_pwr_down

Enable or disable smart power down features on newer adapters. Default 0 (disabled).

hw.em.sbp

Show bad packets when in promiscuous mode. Default 0 (off).

hw.em.rx_int_delay

This value delays the generation of receive interrupts in units of 1.024 microseconds. The default value is 0, since adapters may hang with this feature being enabled.

hw.em.rx_abs_int_delay

If *hw.em.rx_int_delay* is non-zero, this tunable limits the maximum delay in which a receive interrupt is generated.

hw.em.tx_int_delay

This value delays the generation of transmit interrupts in units of 1.024 microseconds. The default value is 64.

hw.em.tx_abs_int_delay

If *hw.em.tx_int_delay* is non-zero, this tunable limits the maximum delay in which a transmit interrupt is generated.

hw.em.max_interrupt_rate

Maximum interrupts per second. The default value is 8000.

hw.em.rx_process_limit

Maximum number of received packets to process at a time, -1 means unlimited. The default value is 100.

FILES

*/dev/led/em** identification LED device nodes

EXAMPLES

Make the identification LED of em0 blink:

```
echo f2 > /dev/led/em0
```

Turn the identification LED of em0 off again:

```
echo 0 > /dev/led/em0
```

DIAGNOSTICS

em%d: Unable to allocate bus resource: memory A fatal initialization error has occurred.

em%d: Unable to allocate bus resource: interrupt A fatal initialization error has occurred.

em%d: watchdog timeout -- resetting The device has stopped responding to the network, or there is a

problem with the network connection (cable).

SUPPORT

For general information and support, go to the Intel support website at: <http://support.intel.com>.

If an issue is identified with the released source code on the supported kernel with a supported adapter, email the specific information related to the issue to [<freebsd@intel.com>](mailto:freebsd@intel.com).

SEE ALSO

[altq\(4\)](#), [arp\(4\)](#), [iflib\(4\)](#), [led\(4\)](#), [netintro\(4\)](#), [ng_ether\(4\)](#), [polling\(4\)](#), [vlan\(4\)](#), [ifconfig\(8\)](#)

HISTORY

The **em** device driver first appeared in FreeBSD 4.4. **em** was merged with the **lem** and **igb** device driver and converted to the **iflib** framework in FreeBSD 12.0.

AUTHORS

The **em** driver was originally written by Intel Corporation [<freebsd@intel.com>](mailto:freebsd@intel.com). It was merged with **igb** driver and converted to the **iflib** framework by Matthew Macy [<mmacy@mattmacy.io>](mailto:mmacy@mattmacy.io) and Sean Bruno [<sbruno@FreeBSD.org>](mailto:sbruno@FreeBSD.org).