

# Linux Base Driver for Intel(R) Ethernet Network Connection

Intel Gigabit Linux driver. Copyright(c) 1999-2018 Intel Corporation.

## Contents

- Identifying Your Adapter
- Command Line Parameters
- Additional Configurations
- Support

## Identifying Your Adapter

For information on how to identify your adapter, and for the latest Intel network drivers, refer to the Intel Support website: <https://www.intel.com/support>

## Command Line Parameters

If the driver is built as a module, the following optional parameters are used by entering them on the command line with the modprobe command using this syntax:

```
modprobe igb [<option>=<VAL1>,<VAL2>,...]
```

There needs to be a <VAL#> for each network port in the system supported by this driver. The values will be applied to each instance, in function order. For example:

```
modprobe igb max_vfs=2,4
```

In this case, there are two network ports supported by igb in the system.

NOTE: A descriptor describes a data buffer and attributes related to the data buffer. This information is accessed by the hardware.

### max\_vfs

**Valid Range:** 0-7

This parameter adds support for SR-IOV. It causes the driver to spawn up to max\_vfs worth of virtual functions. If the value is greater than 0 it will also force the VMDq parameter to be 1 or more.

The parameters for the driver are referenced by position. Thus, if you have a dual port adapter, or more than one adapter in your system, and want N virtual functions per port, you must specify a number for each port with each parameter separated by a comma. For example:

```
modprobe igb max_vfs=4
```

This will spawn 4 VFs on the first port.

```
modprobe igb max_vfs=2,4
```

This will spawn 2 VFs on the first port and 4 VFs on the second port.

NOTE: Caution must be used in loading the driver with these parameters. Depending on your system configuration, number of slots, etc., it is impossible to predict in all cases where the positions would be on the command line.

NOTE: Neither the device nor the driver control how VFs are mapped into config space. Bus layout will vary by operating system. On operating systems that support it, you can check sysfs to find the mapping.

NOTE: When either SR-IOV mode or VMDq mode is enabled, hardware VLAN filtering and VLAN tag stripping/insertion will remain enabled. Please remove the old VLAN filter before the new VLAN filter is added. For example:

```
ip link set eth0 vf 0 vlan 100 // set vlan 100 for VF 0
ip link set eth0 vf 0 vlan 0 // Delete vlan 100
ip link set eth0 vf 0 vlan 200 // set a new vlan 200 for VF 0
```

## Debug

**Valid Range:** 0-16 (0=none,...,16=all)

**Default Value:** 0

This parameter adjusts the level debug messages displayed in the system logs.

## Additional Features and Configurations

### Jumbo Frames

Jumbo Frames support is enabled by changing the Maximum Transmission Unit (MTU) to a value larger than the default value of 1500.

Use the `ifconfig` command to increase the MTU size. For example, enter the following where `<x>` is the interface number:

```
ifconfig eth<x> mtu 9000 up
```

Alternatively, you can use the `ip` command as follows:

```
ip link set mtu 9000 dev eth<x>
ip link set up dev eth<x>
```

This setting is not saved across reboots. The setting change can be made permanent by adding 'MTU=9000' to the file:

- For RHEL: `/etc/sysconfig/network-scripts/ifcfg-eth<x>`
- For SLES: `/etc/sysconfig/network/<config_file>`

NOTE: The maximum MTU setting for Jumbo Frames is 9216. This value coincides with the maximum Jumbo Frames size of 9234 bytes.

NOTE: Using Jumbo frames at 10 or 100 Mbps is not supported and may result in poor performance or loss of link.

### ethtool

The driver utilizes the `ethtool` interface for driver configuration and diagnostics, as well as displaying statistical information. The latest `ethtool` version is required for this functionality. Download it at:

<https://www.kernel.org/pub/software/network/ethtool/>

### Enabling Wake on LAN (WoL)

WoL is configured through the `ethtool` utility.

WoL will be enabled on the system during the next shut down or reboot. For this driver version, in order to enable WoL, the `igb` driver must be loaded prior to shutting down or suspending the system.

NOTE: Wake on LAN is only supported on port A of multi-port devices. Also Wake On LAN is not supported for the following device: - Intel(R) Gigabit VT Quad Port Server Adapter

### Multiqueue

In this mode, a separate MSI-X vector is allocated for each queue and one for "other" interrupts such as link status change and errors. All interrupts are throttled via interrupt moderation. Interrupt moderation must be used to avoid interrupt storms while the driver is processing one interrupt. The moderation value should be at least as large as the expected time for the driver to process an interrupt. Multiqueue is off by default.

REQUIREMENTS: MSI-X support is required for Multiqueue. If MSI-X is not found, the system will fallback to MSI or to Legacy interrupts. This driver supports receive multiqueue on all kernels that support MSI-X.

NOTE: On some kernels a reboot is required to switch between single queue mode and multiqueue mode or vice-versa.

### MAC and VLAN anti-spoofing feature

When a malicious driver attempts to send a spoofed packet, it is dropped by the hardware and not transmitted.

An interrupt is sent to the PF driver notifying it of the spoof attempt. When a spoofed packet is detected, the PF driver will send the following message to the system log (displayed by the `"dmesg"` command): Spoof event(s) detected on VF(n), where n = the VF that attempted to do the spoofing

### Setting MAC Address, VLAN and Rate Limit Using IProute2 Tool

You can set a MAC address of a Virtual Function (VF), a default VLAN and the rate limit using the `IProute2` tool. Download the latest version of the `IProute2` tool from Sourceforge if your version does not have all the features you require.

### Credit Based Shaper (Qav Mode)

When enabling the CBS qdisc in the hardware offload mode, traffic shaping using the CBS (described in the IEEE 802.1Q-2018 Section 8.6.8.2 and discussed in the Annex L) algorithm will run in the i210 controller, so it's more accurate and uses less CPU.

When using offloaded CBS, and the traffic rate obeys the configured rate (doesn't go above it), CBS should have little to no effect in the latency.

The offloaded version of the algorithm has some limits, caused by how the idle slope is expressed in the adapter's registers. It can only represent idle slopes in 16.38431 kbps units, which means that if a idle slope of 2576kbps is requested, the controller will be configured to use a idle slope of ~2589 kbps, because the driver rounds the value up. For more details, see the comments on `:func: igb_config_tx_modes()`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\device\_drivers\ethernet\intel\linux-master)**  
**[Documentation] [networking] [device\_drivers] [ethernet] [intel] igb.rst, line 191); [backlink](#)**  
Unknown interpreted text role "c:func".

NOTE: This feature is exclusive to i210 models.

## Support

For general information, go to the Intel support website at:

<https://www.intel.com/support/>

or the Intel Wired Networking project hosted by Sourceforge at:

<https://sourceforge.net/projects/e1000>

If an issue is identified with the released source code on a supported kernel with a supported adapter, email the specific information related to the issue to [e1000-devel@lists.sf.net](mailto:e1000-devel@lists.sf.net).