

**NAME**

**iavf** - Intel Adaptive Virtual Function driver

**SYNOPSIS**

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

```
device iflib  
device iavf
```

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

```
if_iavf_load="YES"
```

**DESCRIPTION**

The **iavf** driver provides support for the PCI Virtual Functions from the 700 Series of ethernet devices and newer product families. The driver supports Jumbo Frames, TX/RX checksum offload, TCP segmentation offload (TSO), Large Receive Offload (LRO), VLAN tag insertion/extraction, VLAN checksum offload, VLAN TSO, and Receive Side Steering (RSS), all for both IPv4 and IPv6. For further hardware information and questions related to hardware requirements, see <http://support.intel.com/>.

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 9706.

Offloads are also controlled via the interface, for instance, checksumming for both IPv4 and IPv6 can be set and unset, TSO4 and/or TSO6, and finally LRO can be set and unset.

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

*NOTE:* This **iavf** driver is only for Virtual Functions. For 700 series Physical Functions, use the ixl(4) driver.

**LOADER TUNABLES**

Tunables can be set at the loader(8) prompt before booting the kernel or stored in loader.conf(5).

*hw.iavf.rx\_itr*

The RX interrupt rate value, set to 62 (124 usec) by default.

*hw.iavf.tx\_itr*

The TX interrupt rate value, set to 122 (244 usec) by default.

#### *hw.iavf.enable\_head\_writeback*

When the driver is finding the last TX descriptor processed by the hardware, use a value written to memory by the hardware instead of scanning the descriptor ring for completed descriptors. Disabled by default; this mimics the "legacy" TX behavior found in `ixgbe(4)`. to ensure compatibility with future, non-700 series VF devices.

## **SUPPORT**

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

If an issue is identified with this driver with a supported adapter, email all the specific information related to the issue to [frebsd@intel.com](mailto:frebsd@intel.com).

## **SEE ALSO**

`arp(4)`, `ixl(4)`, `netintro(4)`, `vlan(4)`, `ifconfig(8)`, `iflib(9)`

## **HISTORY**

The `iavf` device driver first appeared in FreeBSD 10.1. under the name "ixlv" It was converted to use `iflib(9)` and changed to its current name in FreeBSD 12.

## **AUTHORS**

The `iavf` driver was written by Jack Vogel <[jfv@frebsd.org](mailto:jfv@frebsd.org)> and Eric Joyner <[erj@frebsd.org](mailto:erj@frebsd.org)>.

## **CAVEATS**

This driver is supposed to function on VFs spawned from future network devices by Intel, but at the time of this writing, has only been tested on the 700 series VFs.