

**NAME**

**iwi** - Intel PRO/Wireless 2200BG/2225BG/2915ABG IEEE 802.11 driver

**SYNOPSIS**

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

```
device iwi  
device iwifw  
device pci  
device wlan  
device firmware
```

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

```
if_iwi_load="YES"
```

In both cases, place the following line in loader.conf(5) to acknowledge the firmware license (see below):

```
legal.intel_iwi.license_ack=1
```

**DESCRIPTION**

The **iwi** driver provides support for Intel PRO/Wireless 2200BG/2915ABG MiniPCI and 2225BG PCI network adapters. **iwi** supports **station**, **adhoc**, and **monitor** mode operation. Only one virtual interface may be configured at any time. For more information on configuring this device, see ifconfig(8).

This driver requires the firmware built with the **iwifw** module to work. For the loaded firmware to be enabled for use the license at `/usr/share/doc/legal/intel_iwi.LICENSE` must be agreed by adding the following line to loader.conf(5):

```
legal.intel_iwi.license_ack=1
```

**FILES**

`/usr/share/doc/legal/intel_iwi.LICENSE` **iwi** firmware license

**EXAMPLES**

Join an existing BSS network (i.e., connect to an access point):

```
ifconfig wlan create wlandev iwi0 inet 192.168.0.20 \  
netmask 0xfffff00
```

Join a specific BSS network with network name "my\_net":

```
ifconfig wlan create wlandev iwi0 ssid my_net up
```

Join a specific BSS network with 64-bit WEP encryption:

```
ifconfig wlan create wlandev iwi0 ssid my_net \  
wepmode on wepkey 0x1234567890 weptxkey 1 up
```

Join a specific BSS network with 128-bit WEP encryption:

```
ifconfig wlan create wlandev iwi0 wlanmode adhoc ssid my_net \  
wepmode on wepkey 0x01020304050607080910111213 weptxkey 1
```

## DIAGNOSTICS

**iwi%d: device timeout** The driver will reset the hardware. This should not happen.

**iwi%d: firmware error** The onboard microcontroller crashed for some reason. The driver will reset the hardware. This should not happen.

**iwi%d: timeout waiting for firmware initialization to complete** The onboard microcontroller failed to initialize in time. This should not happen.

**iwi%d: could not load firmware image '%s'** The driver failed to load the firmware image using the `firmware(9)` subsystem. Verify the `iwifw(4)` firmware module is installed and the license agreement loader(8) `tunable` has been set.

**iwi%d: could not load boot firmware** An attempt to upload the boot firmware image to the onboard microcontroller failed. This should not happen.

**iwi%d: could not load microcode** An attempt to upload the microcode image to the onboard microcontroller failed. This should not happen.

**iwi%d: could not load main firmware** An attempt to upload the main firmware image to the onboard microcontroller failed. This should not happen.

## SEE ALSO

`iwifw(4)`, `pci(4)`, `wlan(4)`, `wlan_ccmp(4)`, `wlan_tkip(4)`, `wlan_wep(4)`, `ifconfig(8)`, `wpa_supplicant(8)`

## AUTHORS

The original **iwi** driver was written by Damien Bergamini <[damien.bergamini@free.fr](mailto:damien.bergamini@free.fr)>.