

NAME

iftab – static information about the network interfaces

DESCRIPTION

The file `/etc/iftab` contains descriptive information about the various network interfaces. **iftab** is only used by the program `ifrename(8)` to assign a consistent network interface name to each network interface.

`/etc/iftab` defines a set of *mappings*. Each mapping contains an interface name and a set of selectors. The selectors allow **ifrename** to identify each network interface on the system. If a network interface matches all descriptors of a mapping, **ifrename** attempt to change the name of the interface to the interface name given by the mapping.

MAPPINGS

Each mapping is described on a separate line, it starts with an *interface name*, and contains a set of *descriptors*, separated by space or tabs.

The relationship between descriptors of a mapping is a *logical and*. A mapping matches a network interface only if all the descriptors match. If a network interface doesn't support a specific descriptor, it won't match any mappings using this descriptor.

If you want to use alternate descriptors for an interface name (logical or), specify two different mappings with the same interface name (one on each line). **Ifrename** always use the first matching mapping starting from the *end* of **iftab**, therefore more restrictive mapping should be specified last.

INTERFACE NAME

The first part of each mapping is an interface name. If a network interface matches all descriptors of a mapping, **ifrename** attempt to change the name of the interface to the interface name given by the mapping.

The interface name of a mapping is either a plain interface name (such as *eth2* or *wlan1*) or a interface name pattern containing a single wildcard (such as *eth** or *wlan**). In case of wildcard, the kernel replace the '*' with the lowest available integer making this interface name unique. Note that wildcard is only supported for kernel 2.6.1 and 2.4.30 and later.

It is discouraged to try to map interfaces to default interfaces names such as *eth0*, *wlan0* or *ppp0*. The kernel use those as the default name for any new interface, therefore most likely an interface will already use this name and prevent **ifrename** to use it. Even if you use takeover, the interface may already be up in some cases. Not using those name will allow you to immediately spot unconfigured or new interfaces.

Good names are either totally unique and meaningful, such as *mydsl* or *privatehub*, or use larger integer, such as *eth5* or *wlan5*. The second type is usually easier to integrate in various network utilities.

DESCRIPTORS

Each descriptor is composed of a descriptor name and descriptor value. Descriptors specify a static attribute of a network interface, the goal is to uniquely identify each piece of hardware.

Most users will only use the **mac** selector despite its potential problems, other selectors are for more specialised setup. Most selectors accept a '*' in the selector value for wildcard matching, and most selectors are case insensitive.

mac *mac address*

Matches the MAC Address of the interface with the specified MAC address. The MAC address of the interface can be shown using `ifconfig(8)` or `ip(8)`.

This is the most common selector, as most interfaces have a unique MAC address allowing to identify network interfaces without ambiguity. However, some interfaces don't have a valid MAC address until they are brought up, in such case using this selector is tricky or impossible.

arp *arp type*

Matches the ARP Type (also called Link Type) of the interface with the specified ARP type as a number. The ARP Type of the interface can be shown using `ifconfig(8)` or `ip(8)`, the **link/ether** type correspond to **1** and the **link/ieee802.11** type correspond to **801**.

This selector is useful when a driver create multiple network interfaces for a single network card.

driver *driver name*

Matches the Driver Name of the interface with the specified driver name. The Driver Name of the interface can be shown using *ethtool -i(8)*.

businfo *bus information*

Matches the Bus Information of the interface with the specified bus information. The Bus Information of the interface can be shown using *ethtool -i(8)*.

firmware *firmware revision*

Matches the Firmware Revision of the interface with the firmware revision information. The Firmware Revision of the interface can be shown using *ethtool -i(8)*.

baseaddress *base address*

Matches the Base Address of the interface with the specified base address. The Base Address of the interface can be shown using *ifconfig(8)*.

Because most cards use dynamic allocation of the Base Address, this selector is only useful for ISA and EISA cards.

irq *irq line*

Matches the IRQ Line (interrupt) of the interface with the specified IRQ line. The IRQ Line of the interface can be shown using *ifconfig(8)*.

Because there are IRQ Lines may be shared, this selector is usually not sufficient to uniquely identify an interface.

iwproto *wireless protocol*

Matches the Wireless Protocol of the interface with the specified wireless protocol. The Wireless Protocol of the interface can be shown using *iwconfig(8)* or *iwgetid(8)*.

This selector is only supported on wireless interfaces and is not sufficient to uniquely identify an interface.

pcmciaslot *pcmcia slot*

Matches the Pcmcia Socket number of the interface with the specified slot number. Pcmcia Socket number of the interface can be shown using *cardctl ident(8)*.

This selector is usually only supported on 16 bits cards, for 32 bits cards it is advised to use the selector **businfo**.

prevname *previous interface name*

Matches the name of the interface prior to renaming with the specified oldname.

This selector should be avoided as the previous interface name may vary depending on various condition. A system/kernel/driver update may change the original name. Then, *ifrename* or another tool may rename it prior to the execution of this selector.

SYSFS{filename} *value*

Matches the content the sysfs attribute given by filename to the specified value. For symlinks and parents directories, match the actual directory name of the sysfs attribute given by filename to the specified value.

A list of the most useful sysfs attributes is given in the next section.

SYSFS DESCRIPTORS

Sysfs attributes for a specific interface are located on most systems in the directory named after that interface at */sys/class/net/*. Most sysfs attribute are files, and their values can be read using *cat(1)* or *more(1)*. It is also possible to match attributes in subdirectories.

Some sysfs attributes are symlinks, pointing to another directory in sysfs. If the attribute filename is a symlink the sysfs attribute resolves to the name of the directory pointed by the symlink using *readlink(1)*. The location is a directory in the sysfs tree is also important. If the attribute filename ends with */..*, the sysfs attribute resolves to the real name of the parent directory using *pwd(1)*.

The sysfs filesystem is only supported with 2.6.X kernel and need to be mounted (usually in */sys*). sysfs selectors are not as efficient as other selectors, therefore they should be avoided for maximum performance.

These are common sysfs attributes and their corresponding *ifrename* descriptors.

SYSFS{address} value

Same as the **mac** descriptor.

SYSFS{type} value

Same as the **arp** descriptor.

SYSFS{device} value

Valid only up to kernel 2.6.20. Same as the **businfo** descriptor.

SYSFS{..} value

Valid only from kernel 2.6.21. Same as the **businfo** descriptor.

SYSFS{device/driver} value

Valid only up to kernel 2.6.20. Same as the **driver** descriptor.

SYSFS{../driver} value

Valid only from kernel 2.6.21. Same as the **driver** descriptor.

SYSFS{device/irq} value

Valid only up to kernel 2.6.20. Same as the **irq** descriptor.

SYSFS{../irq} value

Valid only from kernel 2.6.21. Same as the **irq** descriptor.

EXAMPLES

```
# This is a comment
eth2          mac 08:00:09:DE:82:0E
eth3          driver wavelan interrupt 15 baseaddress 0x390
eth4          driver pcnet32 businfo 0000:02:05.0
air*          mac 00:07:0E:* arp 1
myvpn SYSFS{address} 00:10:83:* SYSFS{type} 1
bcm*          SYSFS{device} 0000:03:00.0 SYSFS{device/driver} bcm43xx
bcm*          SYSFS{..} 0000:03:00.0 SYSFS{../driver} bcm43xx
```

AUTHOR

Jean Tourrilhes – jt@hpl.hp.com

FILES

/etc/iftab

SEE ALSO

ifrename(8), ifconfig(8), ip(8), ethtool(8), iwconfig(8).