

---

# interfaces

Fichier de configuration pour ifup et ifdown. Il contient les informations de configuration pour les interfaces réseau

Les lignes commençant par "#" sont ignorées. Une ligne peut être étendue sur plusieurs en utilisant un "\n" à la fin de la ligne. Ce fichier consiste d'une ou plusieurs "iface", "mapping", "auto" et "allow-".

Les lignes commençant par le mot "auto" sont utilisées pour identifier les interfaces physiques à configurer quand ifup est lancé avec l'option -a (cette options est utilisée par le système au démarrage). Le nom de l'interface physique doit suivre le mot "auto" sur la même ligne.

Les lignes commençant avec "allow-" sont utilisées pour identifier les interfaces qui devraient être activées automatiquement par divers sous-systèmes. Cela peut-être fait avec une commande du type "ifup -allow=hotplug eth0 eth1", qui va activer uniquement eth0 et eth1 s'ils sont listés dans une ligne "allow-hotplug". Noter que "allow-auto" est synonyme de "auto".

Les lignes commençant pas "mapping" sont utilisés pour déterminer comment le nom d'interface logique est choisis pour une interface physique. chaque mapping doit contenir une définition script.

**ifup** utilise le nom de l'interface physique comme nom logique, à moins qu'un suffix du type =LOGICAL soit spécifié.

La définition des interfaces logique commence avec une ligne consistant du mot "iface", suivi du nom de l'interface logique. Le nom de l'interface est suivi du nom de famille d'adresse que l'interface utilise. cela sera "inet" pour TCP/IP, ipx pour les réseau ipx et inet6 pour les réseaux ipv6. La suite est le nom de la méthode pour configurer l'interface.

Des options additionnelles peuvent être utilisées sur les lignes suivante. ces options dépendent de la méthode et de la famille. Par exemple, le paquet **wireless-tools** ajoute des options préfixés avec "wireless" qui peuvent-être utilisées pour configurer l'interface en utilisant **iwconfig**.

## IFACE options

Ces options sont disponible pour toutes les familles et méthodes. Chacune peuvent être indiqué plusieurs fois et sont exécutés dans l'ordre qu'elle apparaissent. (pour s'assurer qu'une commande ne plante jamais, suffixer par "|| true")

**pre-up command** Lance la commande avant de configurer l'interface.

**up comand**

**post-up command** Lance la commande après avoir configuré l'interface

**down command**

**pre-down command** Lance la commande avant de dé-configurer l'interface.

**post-down** Lance la commande après avoir dé-configuré l'interface.

Il existe pour chacune des options mentionnées, un dossier **/etc/network/if-<option>.d/**. Les scripts qui s'y trouvent sont lancés en utilisant **run-parts** après que l'option ait été traitée.

Toutes ces commandes ont accès aux variables d'environnement suivante :

**IFACE** Nom physique de l'interface

**LOGICAL** Nom logique de l'interface

---

**ADDRFAM** famille de l'interface

**METHOD** Méthode de l'interface

**MODE** Démarre si lancé depuis ifup, arrête si lancé depuis ifdown

**PHASE** Comme pour MODE, mais plus précis, distingue pre-up, post-up, pre-down et post-down.

**VERBOSITY** indique si `-verbose` est utilisé.

**PATH** le path de recherche de commande : `/usr/local/sbin` `/usr/local/bin` `/usr/sbin` `/usr/bin` `/sbin` `/bin`

## Famille d'adresse inet

**la méthode loopback** Cette méthode est utilisé pour définir une interface de bouclage ipv4. N'a pas d'options

**Méthode static** Cette méthode est utilisée pour définir les interfaces ethernet avec des adresses ipv4 définie statiquement dans `/etc/network/interfaces`

## Options

**address** adresse IP

**netmask** masque de sous-réseau

**broadcast** adresse de broadcast

**network** adresse du réseau

**metric** métrique pour la route par défaut

**gateway** passerelle par défaut

**pointopoint** adresse de "l'autre bout"

**media** dépendant du pilote.

**hwaddress** classe et adresse physique. classe peut être ether, ax25, ARCnet ou netrom.

**mtu** taille MTU

**La méthode manual** Cette méthode peut être utilisé pour définir des interfaces sans configuration par défaut. N'a pas d'options

**La méthode dhcp** Cette méthode peut être utilisée pour obtenir une adresse et une configuration via DHCP.

## Options

**hostname** nom d'hôte

**leasetime** lease time en seconde

**vendor** Vendor Class Identifier

**Client** identifiant client

**hwaddress** classe et adresse physique.

**La méthode bootp** Cette méthode peut être utilisée pour obtenir une ip via bootp

## Options

**bootfile** fichier à utiliser

**server** adresse du serveur

**hwaddr** utilise l'adresse comme adresse physique

---

**la méthode ppp** Cette méthode est utilisé pour configurer des interfaces ppp (utilise pon/poff)

## Options

**provider** utilise ce nom comme provider

**la méthode wvdial** méthode utilise wvdial pour configurer une interface ppp.

**La méthode ipv4all** cette méthode utilise avahi-autoipd pour configurer une interface avec une adresse ipv4 APIPA (famille 169.254.0.0/16). N'a pas d'options

## Famille d'adresse inet6

**La méthode loopback** Cette méthode peut être utilisée pour définir l'interface de bouclage ipv6. N'a pas d'options

**La méthode static** Utilisé pour définir statiquement l'interface.

## Options

**address** adresse de l'interface

**netmask** masque de sous-réseaux

**gateway** passerelle par défaut

**media** dépendant du pilote

**hwaddress** classe et adresse physique. Classe peut être ether, ARCnet ou netrom

**mtu** taille MTU

**La méthode manual** Cette méthode est utilisé pour définir des interfaces pour lesquelles aucune configuration n'est faite. N'a pas d'options

**La méthode v4tunnel** Cette méthode peut être utilisée pour créer un tunnel IPv6-over-IPv4. Il requière la commande ip du paquet iproute.

## Options

**address** adresse de l'interface

**netmask** masque de sous-réseaux

**endpoint** adresse du bout du tunnel (ipv4)

**local** adresse local du tunnel (ipv4)

**gateway** passerelle par défaut

**ttl** réglage ttl

## Exemples

interface loopback :

```
auto lo
```

```
iface lo inet loopback
```

configuration basique d'une carte ethernet :

---

```
auto eth0
iface eth0 inet static
address 192.168.0.42
network 192.168.0.0
netmask 255.255.255.0
gateway 192.168.0.1
```

configuration plus complexe, avec mise a jour de la table de routage :

```
auto eth0
iface eth0 inet static
address 192.168.1.42
network 192.168.1.0
netmask 255.255.255.128
broadcast 192.168.1.0
up route add -net 192.168.1.128 netmask 255.255.255.128 gw 192.168.1.2
up route add default gw 192.168.1.200
down route del default gw 192.168.1.200
down route del -net 192.168.1.128 netmask 255.255.255.128 gw 192.168.1.2
```

configuration une carte ethernet avec 2 interfaces :

```
auto eth0 eth0 :1
iface eth0 inet static
address 192.168.0.100
network 192.168.0.0
netmask 255.255.255.0
broadcast 192.168.0.255
gateway 192.168.0.1
iface eth0 :1 inet static
address 192.168.0.200
network 192.168.0.0
netmask 255.255.255.0
```