NAME

systemd.link − Network device configuration

SYNOPSIS

link.link

DESCRIPTION

Network link configuration is performed by the **net_setup_link** udev builtin.

The link files are read from the files located in the system network directory /lib/systemd/network, the volatile runtime network directory /run/systemd/network, and the local administration network directory /etc/systemd/network. Link files must have the extension .link; other extensions are ignored. All link files are collectively sorted and processed in lexical order, regardless of the directories in which they live. However, files with identical filenames replace each other. Files in /etc have the highest priority, files in /run take precedence over files with the same name in /lib. This can be used to override a system−supplied link file with a local file if needed. As a special case, an empty file (file size 0) or symlink with the same name pointing to /dev/null disables the configuration file entirely (it is "masked").

The link file contains a [Match] section, which determines if a given link file may be applied to a given device, as well as a [Link] section specifying how the device should be configured. The first (in lexical order) of the link files that matches a given device is applied. Note that a default file 99−default.link is shipped by the system. Any user−supplied .link should hence have a lexically earlier name to be considered at all.

See **udevadm**(8) for diagnosing problems with .link files.

[MATCH] SECTION OPTIONS

A link file is said to match a device if each of the entries in the [Match] section matches, or if the section is empty. The following keys are accepted:

MACAddress=

A whitespace−separated list of hardware addresses. Use full colon−, hyphen− or dot−delimited hexadecimal. See the example below. This option may appear more than once, in which case the lists are merged. If the empty string is assigned to this option, the list of hardware addresses defined prior to this is reset.

Example:

```
MACAddress=01:23:45:67:89:ab 00−11−22−33−44−55 AABB.CCDD.EEFF
```

OriginalName=

A whitespace−separated list of shell−style globs matching the device name, as exposed by the udev property "INTERFACE". This cannot be used to match on names that have already been changed from userspace. Caution is advised when matching on kernel−assigned names, as they are known to be unstable between reboots.

Path=

A whitespace−separated list of shell−style globs matching the persistent path, as exposed by the udev property *ID_PATH*.

Driver=

A whitespace−separated list of shell−style globs matching the driver currently bound to the device, as exposed by the udev property *DRIVER* of its parent device, or if that is not set, the driver as exposed by **ethtool −i** of the device itself.

Type=

A whitespace−separated list of shell−style globs matching the device type, as exposed by the udev property *DEVTYPE*.

Host=

Matches against the hostname or machine ID of the host. See *ConditionHost=* in **systemd.unit**(5) for details. When prefixed with an exclamation mark ("!"), the result is negated. If an empty string is assigned, then previously assigned value is cleared.

Virtualization=
Checks whether the system is executed in a virtualized environment and optionally test whether it is a specific implementation. See *ConditionVirtualization=* in **systemd.unit**(5) for details. When prefixed with an exclamation mark ("!"), the result is negated. If an empty string is assigned, then previously assigned value is cleared.

KernelCommandLine=
Checks whether a specific kernel command line option is set. See *ConditionKernelCommandLine=* in **systemd.unit**(5) for details. When prefixed with an exclamation mark ("!"), the result is negated. If an empty string is assigned, then previously assigned value is cleared.

KernelVersion=
Checks whether the kernel version (as reported by **uname −r**) matches a certain expression. See *ConditionKernelVersion=* in **systemd.unit**(5) for details. When prefixed with an exclamation mark ("!"), the result is negated. If an empty string is assigned, then previously assigned value is cleared.

Architecture=
Checks whether the system is running on a specific architecture. See *ConditionArchitecture=* in **systemd.unit**(5) for details. When prefixed with an exclamation mark ("!"), the result is negated. If an empty string is assigned, then previously assigned value is cleared.

[LINK] SECTION OPTIONS

The [Link] section accepts the following keys:

Description=
A description of the device.

Alias=
The *ifalias* interface property is set to this value.

MACAddressPolicy=
The policy by which the MAC address should be set. The available policies are:

persistent
If the hardware has a persistent MAC address, as most hardware should, and if it is used by the kernel, nothing is done. Otherwise, a new MAC address is generated which is guaranteed to be the same on every boot for the given machine and the given device, but which is otherwise random. This feature depends on ID_NET_NAME_* properties to exist for the link. On hardware where these properties are not set, the generation of a persistent MAC address will fail.

random
If the kernel is using a random MAC address, nothing is done. Otherwise, a new address is randomly generated each time the device appears, typically at boot. Either way, the random address will have the "unicast" and "locally administered" bits set.

none
Keeps the MAC address assigned by the kernel.

MACAddress=
The MAC address to use, if no *MACAddressPolicy=* is specified.

NamePolicy=
An ordered, space−separated list of policies by which the interface name should be set. *NamePolicy=* may be disabled by specifying **net.ifnames=0** on the kernel command line. Each of the policies may fail, and the first successful one is used. The name is not set directly, but is exported to udev as the property **ID_NET_NAME**, which is, by default, used by a **udev**(7), rule to set *NAME*. The available policies are:

kernel
If the kernel claims that the name it has set for a device is predictable, then no renaming is performed.

database
: The name is set based on entries in the udev's Hardware Database with the key *ID_NET_NAME_FROM_DATABASE*.

onboard
: The name is set based on information given by the firmware for on−board devices, as exported by the udev property *ID_NET_NAME_ONBOARD*.

slot
: The name is set based on information given by the firmware for hot−plug devices, as exported by the udev property *ID_NET_NAME_SLOT*.

path
: The name is set based on the device's physical location, as exported by the udev property *ID_NET_NAME_PATH*.

mac
: The name is set based on the device's persistent MAC address, as exported by the udev property *ID_NET_NAME_MAC*.

keep
: If the device already had a name given by userspace (as part of creation of the device or a rename), keep it.

Name=

The interface name to use. This option has lower precedence than *NamePolicy=*, so for this setting to take effect, *NamePolicy=* must either be unset, empty, disabled, or all policies configured there must fail. Also see the example below with "Name=dmz0".

Note that specifying a name that the kernel might use for another interface (for example "eth0") is dangerous because the name assignment done by udev will race with the assignment done by the kernel, and only one interface may use the name. Depending on the order of operations, either udev or the kernel will win, making the naming unpredictable. It is best to use some different prefix, for example "internal0"/"external0" or "lan0"/"lan1"/"lan3".

MTUBytes=

The maximum transmission unit in bytes to set for the device. The usual suffixes K, M, G, are supported and are understood to the base of 1024.

BitsPerSecond=

The speed to set for the device, the value is rounded down to the nearest Mbps. The usual suffixes K, M, G, are supported and are understood to the base of 1000.

Duplex=

The duplex mode to set for the device. The accepted values are **half** and **full**.

AutoNegotiation=

Takes a boolean. If set to yes, automatic negotiation of transmission parameters is enabled. Autonegotiation is a procedure by which two connected ethernet devices choose common transmission parameters, such as speed, duplex mode, and flow control. When unset, the kernel's default will be used.

Note that if autonegotiation is enabled, speed and duplex settings are read−only. If autonegotation is disabled, speed and duplex settings are writable if the driver supports multiple link modes.

WakeOnLan=

The Wake−on−LAN policy to set for the device. The supported values are:

phy
: Wake on PHY activity.

unicast

Wake on unicast messages.

multicast
: Wake on multicast messages.

broadcast
: Wake on broadcast messages.

arp
: Wake on ARP.

magic
: Wake on receipt of a magic packet.

secureon
: Enable secureon(tm) password for MagicPacket(tm).

off
: Never wake.

Defaults to **off**.

Port=

The port option is used to select the device port. The supported values are:

tp
: An Ethernet interface using Twisted−Pair cable as the medium.

aui
: Attachment Unit Interface (AUI). Normally used with hubs.

bnc
: An Ethernet interface using BNC connectors and co−axial cable.

mii
: An Ethernet interface using a Media Independent Interface (MII).

fibre
: An Ethernet interface using Optical Fibre as the medium.

Advertise=

This sets what speeds and duplex modes of operation are advertised for auto−negotiation. This implies "AutoNegotiation=yes". The supported values are:

Table 1. Supported advertise values

Advertise	**Speed (Mbps)**	**Duplex Mode**
10baset−half	10	half
10baset−full	10	full
100baset−half	100	half
100baset−full	100	full
1000baset−half	1000	half
1000baset−full	1000	full
10000baset−full	10000	full
2500basex−full	2500	full
1000basekx−full	1000	full
10000basekx4−full	10000	full
10000basekr−full	10000	full
10000baser−fec	10000	full
20000basemld2−full	20000	full
20000basekr2−full	20000	full

By default this is unset, i.e. all possible modes will be advertised. This option may be specified more than once, in which case all specified speeds and modes are advertised. If the empty string is assigned to this option, the list is reset, and all prior assignments have no effect.

TCPSegmentationOffload=
Takes a boolean. If set to true, the TCP Segmentation Offload (TSO) is enabled. When unset, the kernel's default will be used.

TCP6SegmentationOffload=
Takes a boolean. If set to true, the TCP6 Segmentation Offload (tx−tcp6−segmentation) is enabled. When unset, the kernel's default will be used.

GenericSegmentationOffload=
Takes a boolean. If set to true, the Generic Segmentation Offload (GSO) is enabled. When unset, the kernel's default will be used.

GenericReceiveOffload=
Takes a boolean. If set to true, the Generic Receive Offload (GRO) is enabled. When unset, the kernel's default will be used.

LargeReceiveOffload=
Takes a boolean. If set to true, the Large Receive Offload (LRO) is enabled. When unset, the kernel's default will be used.

RxChannels=
Sets the number of receive channels (a number between 1 and 4294967295) .

TxChannels=
Sets the number of transmit channels (a number between 1 and 4294967295).

OtherChannels=
Sets the number of other channels (a number between 1 and 4294967295).

CombinedChannels=
Sets the number of combined set channels (a number between 1 and 4294967295).

EXAMPLES

Example 1. /lib/systemd/network/99−default.link

The link file 99−default.link that is shipped with systemd defines the default naming policy for links.

```
[Link]
NamePolicy=kernel database onboard slot path
```

```
MACAddressPolicy=persistent
```

Example 2. /etc/systemd/network/10−dmz.link

This example assigns the fixed name "dmz0" to the interface with the MAC address 00:a0:de:63:7a:e6:

```
[Match]
MACAddress=00:a0:de:63:7a:e6

[Link]
Name=dmz0
```

NamePolicy= is not set, so *Name=* takes effect. We use the "10−" prefix to order this file early in the list. Note that it needs to before "99−link", i.e. it needs a numerical prefix, to have any effect at all.

Example 3. Debugging *NamePolicy=* assignments

```
$ sudo SYSTEMD_LOG_LEVEL=debug udevadm test−builtin net_setup_link /sys/class/net/hub0
...
Parsed configuration file /lib/systemd/network/99−default.link
Parsed configuration file /etc/systemd/network/10−eth0.link
ID_NET_DRIVER=cdc_ether
Config file /etc/systemd/network/10−eth0.link applies to device hub0
link_config: autonegotiation is unset or enabled, the speed and duplex are not writable.
hub0: Device has name_assign_type=4
Using default interface naming scheme 'v240'.
hub0: Policies didn't yield a name, using specified Name=hub0.
ID_NET_LINK_FILE=/etc/systemd/network/10−eth0.link
ID_NET_NAME=hub0
...
```

Explicit *Name=* configuration wins in this case.

```
sudo SYSTEMD_LOG_LEVEL=debug udevadm test−builtin net_setup_link /sys/class/net/enp0s31f6
...
Parsed configuration file /lib/systemd/network/99−default.link
Parsed configuration file /etc/systemd/network/10−eth0.link
Created link configuration context.
ID_NET_DRIVER=e1000e
Config file /lib/systemd/network/99−default.link applies to device enp0s31f6
link_config: autonegotiation is unset or enabled, the speed and duplex are not writable.
enp0s31f6: Device has name_assign_type=4
Using default interface naming scheme 'v240'.
enp0s31f6: Policy *keep*: keeping existing userspace name
enp0s31f6: Device has addr_assign_type=0
enp0s31f6: MAC on the device already matches policy *persistent*
ID_NET_LINK_FILE=/lib/systemd/network/99−default.link
...
```

In this case, the interface was already renamed, so the **keep** policy specified as the first option in 99−default.link means that the existing name is preserved. If **keep** was removed, or if were in boot before the renaming has happened, we might get the following instead:

```
enp0s31f6: Policy *path* yields "enp0s31f6".
enp0s31f6: Device has addr_assign_type=0
enp0s31f6: MAC on the device already matches policy *persistent*
ID_NET_LINK_FILE=/lib/systemd/network/99−default.link
ID_NET_NAME=enp0s31f6
```

```
...
```

Please note that the details of output are subject to change.

Example 4. /etc/systemd/network/10−internet.link

This example assigns the fixed name "internet0" to the interface with the device path "pci−0000:00:1a.0−*":

```
[Match]
Path=pci−0000:00:1a.0−*

[Link]
Name=internet0
```

Example 5. /etc/systemd/network/25−wireless.link

Here's an overly complex example that shows the use of a large number of [Match] and [Link] settings.

```
[Match]
MACAddress=12:34:56:78:9a:bc
Driver=brcmsmac
Path=pci−0000:02:00.0−*
Type=wlan
Virtualization=no
Host=my−laptop
Architecture=x86−64

[Link]
Name=wireless0
MTUBytes=1450
BitsPerSecond=10M
WakeOnLan=magic
MACAddress=cb:a9:87:65:43:21
```

SEE ALSO

systemd-udevd.service(8), **udevadm**(8), **systemd.netdev**(5), **systemd.network**(5)