

**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*=

The hardware address.

*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.

*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.

*KernelCommandLine*=

Checks whether a specific kernel command line option is set (or if prefixed with the exclamation mark unset). See "ConditionKernelCommandLine=" in [systemd.unit\(5\)](#) for details.

*Architecture=*

Checks whether the system is running on a specific architecture. See "ConditionArchitecture=" in [systemd.unit\(5\)](#) for details.

**[LINK] SECTION OPTIONS**

The "[Link]" section accepts the following keys:

*Description=*

A description of the device.

*Alias=*

The "ifalias" 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 rule to set "NAME". If the name has already been set by userspace, no renaming is performed. 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".

*Name*=

The interface name to use in case all the policies specified in *NamePolicy*= fail, or in case *NamePolicy*= is missing or disabled.

*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".

*WakeOnLan*=

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

"phy"

Wake on PHY activity.

"magic"

Wake on receipt of a magic packet.

"off"

Never wake.

*TCPSegmentationOffload*=

The TCP Segmentation Offload (TSO) when true enables TCP segmentation offload. Takes a boolean value. Defaults to "unset".

*GenericSegmentationOffload*=

The Generic Segmentation Offload (GSO) when true enables generic segmentation offload. Takes a boolean value. Defaults to "unset".

*UDPSegmentationOffload*=

The UDP Segmentation Offload (USO) when true enables UDP segmentation offload. Takes a boolean value. Defaults to "unset".

*GenericReceiveOffload*=

The Generic Receive Offload (GRO) when true enables generic receive offload. Takes a boolean value. Defaults to "unset".

*LargeReceiveOffload*=

The Large Receive Offload (LRO) when true enables large receive offload. Takes a boolean value. Defaults to "unset".

## 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
```

**Example 3. /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 4. /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\(5\)](#)