

NAME

ifconfig – configure network interface parameters

SYNOPSIS

```
ifconfig interface [address_family] [address [/prefix_length]
[dest_address]] [addif address [/prefix_length]]
[removeif address [/prefix_length]] [arp | -arp]
[auth_algs authentication_algorithm] [encr_algs encryption_algorithm]
[encr_auth_algs authentication_algorithm] [auto-revarp]
[broadcast address] [deprecated | -deprecated]
[preferred | -preferred] [destination dest_address]
[ether [address]] [failover | -failover] [group
[name | ""]] [index if_index] [metric n] [modlist]
[modinsert mod_name@pos] [modremove mod_name@pos]
[mtu n] [netmask mask] [plumb] [unplumb] [private
| -private] [nud | -nud] [set [address] [/netmask]]
[standby | -standby] [subnet subnet_address] [tdst
tunnel_dest_address] [token address/prefix_length]
[tsrc tunnel_src_address] [trailers | -trailers]
[up] [down] [usesrc [name | none]] [xmit | -xmit]
[encaplimit n | -encaplimit] [thoplimit n] [router
| -router] [zone zonename | -zone | -all-zones]
```

```
ifconfig [address_family] interface {auto-dhcp | dhcp} [primary]
[wait seconds] [drop | extend | inform | ping
| release | start | status]
```

DESCRIPTION

The command **ifconfig** is used to assign an address to a network interface and to configure network interface parameters. The **ifconfig** command must be used at boot time to define the network address of each interface present on a machine; it may also be used at a later time to redefine an interface's address or other operating parameters. If no option is specified, **ifconfig** displays the current configuration for a network interface. If an address family is specified, **ifconfig** reports only the details specific to that address family. Only privileged users may modify the configuration of a network interface. Options appearing within braces ({ }) indicate that one of the options must be specified.

DHCP Configuration

The forms of **ifconfig** that use the **auto-dhcp** or **dhcp** arguments are used to control the Dynamic Host Configuration Protocol ("DHCP") configuration of the interface. In this mode, **ifconfig** is used to control operation of **dhcagent**(1M), the DHCP client daemon. Once an interface is placed under DHCP control by using the **start** operand, **ifconfig** should not, in normal operation, be used to modify the address or characteristics of the interface. If the address of an interface under DHCP is changed, **dhcagent** will remove the interface from its control.

OPTIONS

The following options are supported:

addif *address*

Create the next unused logical interface on the specified physical interface. If the physical interface is part of a multipathing group, the logical interface can be added to a different physical interface in the same group.

all-zones

Make the interface available to every shared-IP zone on the system. The appropriate zone to which to deliver data is determined using the **tnzonecfg** database. This option is available only if the system is configured with the Solaris Trusted Extensions feature.

The **tnzonecfg** database is described in the **tnzonecfg(4)** man page, which is part of the *Solaris Trusted Extensions Reference Manual*.

anycast

Marks the logical interface as an anycast address by setting the **ANYCAST** flag. See "INTERFACE FLAGS," below, for more information on anycast.

-anycast

Marks the logical interface as not an anycast address by clearing the **ANYCAST** flag.

arp

Enable the use of the Address Resolution Protocol ("**ARP**") in mapping between network level addresses and link level addresses (default). This is currently implemented for mapping between IPv4 addresses and MAC addresses.

-arp

Disable the use of the **ARP** on a physical interface.

auth_algs *authentication algorithm*

For a tunnel, enable IPsec **AH** with the authentication algorithm specified. The algorithm can be either a number or an algorithm name, including *any* to express no preference in algorithm. All IPsec tunnel properties must be specified on the same command line. To disable tunnel security, specify an **auth_alg** of **none**.

It is now preferable to use the **ipseconf(1M)** command when configuring a tunnel's security properties. If **ipseconf** was used to set a tunnel's security properties, this keyword will not affect the tunnel.

auto-dhcp

Use DHCP to automatically acquire an address for this interface. This option has a completely equivalent alias called **dhcp**.

For IPv6, the interface specified must be the zeroth logical interface (the physical interface name), which has the link-local address.

primary

Defines the interface as the **primary**. The interface is defined as the preferred one for the delivery of client-wide configuration data. Only one interface can be the primary at any given

time. If another interface is subsequently selected as the primary, it replaces the previous one. Nominating an interface as the primary one will not have much significance once the client work station has booted, as many applications will already have started and been configured with data read from the previous primary interface.

wait *seconds*

The **ifconfig** command will wait until the operation either completes or for the interval specified, whichever is the sooner. If no wait interval is given, and the operation is one that cannot complete immediately, **ifconfig** will wait 30 seconds for the requested operation to complete. The symbolic value **forever** may be used as well, with obvious meaning.

drop

Remove the specified interface from **DHCP** control without notifying the DHCP server, and record the current lease for later use. Additionally, for IPv4, set the IP address to zero and mark the interface as "down." For IPv6, unplumb all logical interfaces plumbed by **dhcagent**.

extend

Attempt to extend the lease on the interface's IP address. This is not required, as the agent will automatically extend the lease well before it expires.

inform

Obtain network configuration parameters from **DHCP** without obtaining a lease on **IP** addresses. This is useful in situations where an **IP** address is obtained through mechanisms other than **DHCP**.

ping

Check whether the interface given is under **DHCP** control, which means that the interface is managed by the **DHCP** agent and is working properly. An exit status of **0** means success.

release

Relinquish the IP addresses on the interface by notifying the server and discard the current lease. For IPv4, mark the interface as "down." For IPv6, all logical interfaces plumbed by **dhcagent** are unplumbed.

start

Start **DHCP** on the interface.

status

Display the **DHCP** configuration status of the interface.

auto-revarp

Use the Reverse Address Resolution Protocol (RARP) to automatically acquire an address for this interface. This will fail if the interface does not support RARP; for example, IPoIB (IP over InfiniBand), and on IPv6 interfaces.

broadcast *address*

For IPv4 only. Specify the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1's. A "+" (plus sign) given for the broadcast value causes the broadcast address to be reset to a default appropriate for the (possibly new) address and netmask. The arguments of **ifconfig** are interpreted left to right. Therefore

```
example% ifconfig -a netmask + broadcast +
```

and

```
example% ifconfig -a broadcast + netmask +
```

may result in different values being assigned for the broadcast addresses of the interfaces.

deprecated

Marks the logical interface as deprecated. An address associated with a deprecated interface will not be used as source address for outbound packets unless either there are no other addresses available on the interface or the application has bound to this address explicitly. The status display shows **DEPRECATED** as part of flags. See for information on the flags supported by **ifconfig**.

-deprecated

Marks a logical interface as not deprecated. An address associated with such an interface could be used as a source address for outbound packets.

preferred

Marks the logical interface as preferred. This option is only valid for IPv6 addresses. Addresses assigned to preferred logical interfaces are preferred as source addresses over all other addresses configured on the system, unless the address is of an inappropriate scope relative to the destination address. Preferred addresses are used as source addresses regardless of which physical interface they are assigned to. For example, you can configure a preferred source address on the loopback interface and advertise reachability of this address by using a routing protocol.

-preferred

Marks the logical interface as not preferred.

destination *dest_address*

Set the destination address for a point-to point interface.

dhcp

This option is an alias for option **auto-dhcp**

down

Mark a logical interface as "down". (That is, turn off the **IFF_UP** bit.) When a logical interface is marked "down," the system does not attempt to use the address assigned to that interface as a source address for outbound packets and will not recognize inbound packets destined to that address as being addressed to this host. Additionally, when all logical interfaces on a given physical interface are "down," the physical interface itself is disabled.

When a logical interface is down, all routes that specify that interface as the output (using the **-ifp** option in the **route(1M)** command or **RTA_IFP** in a **route(7P)** socket) are removed from the forwarding table. Routes marked with **RTF_STATIC** are returned to the table if the interface is brought back up, while routes not marked with **RTF_STATIC** are simply deleted.

When all logical interfaces that could possibly be used to reach a particular gateway address are brought down (specified without the interface option as in the previous paragraph), the affected gateway routes are treated as though they had the **RTF_BLACKHOLE** flag set. All matching packets are discarded because the gateway is unreachable.

encplimit *n*

Set the tunnel encapsulation limit for the interface to *n*. This option applies to IPv4-in-IPv6 and IPv6-in-IPv6 tunnels only. The tunnel encapsulation limit controls how many more tunnels a packet may enter before it leaves any tunnels, that is, the tunnel nesting level.

-encplimit

Disable generation of the tunnel encapsulation limit. This option applies only to IPv4-in-IPv6 and IPv6-in-IPv6 tunnels.

encr_auth_algs *authentication algorithm*

For a tunnel, enable IPsec **ESP** with the authentication algorithm specified. It can be either a number or an algorithm name, including **any** or **none**, to indicate no algorithm preference. If an **ESP** encryption algorithm is specified but the authentication algorithm is not, the default value for the **ESP** authentication algorithm will be **any**.

It is now preferable to use the **ipseconf(1M)** command when configuring a tunnel's security properties. If **ipseconf** was used to set a tunnel's security properties, this keyword will not affect the tunnel.

encr_algs *encryption algorithm*

For a tunnel, enable IPsec **ESP** with the encryption algorithm specified. It can be either a number or an algorithm name. Note that all IPsec tunnel properties must be specified on the same command line. To disable tunnel security, specify the value of **encr_alg** as **none**. If an **ESP** authentication algorithm is specified, but the encryption algorithm is not, the default value for the **ESP** encryption will be **null**.

It is now preferable to use the **ipseccnf(1M)** command when configuring a tunnel's security properties. If **ipseccnf** was used to set a tunnel's security properties, this keyword will not affect the tunnel.

ether [*address*]

If no address is given and the user is root or has sufficient privileges to open the underlying device, then display the current Ethernet address information.

Otherwise, if the user is root or has sufficient privileges, set the Ethernet address of the interfaces to *address*. The address is an Ethernet address represented as *x:x:x:x:x:x* where *x* is a hexadecimal number between 0 and FF. Similarly, for the IPoIB (IP over InfiniBand) interfaces, the address will be 20 bytes of colon-separated hex numbers between **0** and **FF**.

Some, though not all, Ethernet interface cards have their own addresses. To use cards that do not have their own addresses, refer to section 3.2.3(4) of the IEEE 802.3 specification for a definition of the locally administered address space. The use of multipathing groups should be restricted to those cards with their own addresses (see **MULTIPATHING GROUPS**).

-failover

Mark the logical interface as a non-failover interface. Addresses assigned to non-failover logical interfaces will not failover when the interface fails. Status display shows **NOFAILOVER** as part of flags.

failover

Mark the logical interface as a failover interface. An address assigned to such an interface will failover when the interface fails. Status display does not show **NOFAILOVER** as part of flags.

group [*name* | ""]

Insert the logical interface in the multipathing group specified by *name*. To delete an interface from a group, use a null string "". When invoked on the logical interface with id zero, the status display shows the group name.

index *n*

Change the interface index for the interface. The value of *n* must be an interface index (*if_index*) that is not used on another interface. *if_index* will be a non-zero positive number that uniquely identifies the network interface on the system.

metric *n*

Set the routing metric of the interface to *n*; if no value is specified, the default is **0**. The routing metric is used by the routing protocol. Higher metrics have the effect of making a route less favorable. Metrics are counted as addition hops to the destination network or host.

modinsert *mod_name@pos*

Insert a module with name *mod_name* to the stream of the device at position *pos*. The position is relative to the stream head. Position **0** means directly under stream head.

Based upon the example in the **modlist** option, use the following command to insert a module with name **ipqos** under the **ip** module and above the firewall module:

```
example% ifconfig eri0 modinsert ipqos@2
```

A subsequent listing of all the modules in the stream of the device follows:

```
example% ifconfig eri0 modlist
0 arp
1 ip
2 ipqos
3 firewall
4 eri
```

modlist

List all the modules in the stream of the device.

The following example lists all the modules in the stream of the device:

```
example% ifconfig eri0 modlist
0 arp
1 ip
2 firewall
4 eri
```

modremove *mod_name@pos*

Remove a module with name *mod_name* from the stream of the device at position *pos*. The position is relative to the stream head.

Based upon the example in the **modinsert** option, use the following command to remove the firewall module from the stream after inserting the **ipqos** module:

```
example% ifconfig eri0 modremove firewall@3
```

A subsequent listing of all the modules in the stream of the device follows:

```
example% ifconfig eri0 modlist
0 arp
1 ip
2 ipqos
3 eri
```

Note that the core IP stack modules, for example, **ip** and **tun** modules, cannot be removed.

mtu *n*

Set the maximum transmission unit of the interface to *n*. For many types of networks, the **mtu** has an upper limit, for example, **1500** for Ethernet. This option sets the **FIXEDMTU** flag on the affected interface.

netmask *mask*

For IPv4 only. Specify how much of the address to reserve for subdividing networks into subnetworks. The mask includes the network part of the local address and the subnet part, which is taken from the host field of the address. The mask contains 1's for the bit positions in the 32-bit address which are to be used for the network and subnet parts, and 0's for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the network portion. The mask can be specified in one of four ways:

1. with a single hexadecimal number with a leading 0x,
2. with a dot-notation address,
3. with a "+" (plus sign) address, or
4. with a pseudo host name/pseudo network name found in the network database **networks(4)**.

If a "+" (plus sign) is given for the netmask value, the mask is looked up in the **netmasks(4)** database. This lookup finds the longest matching netmask in the database by starting with the interface's IPv4 address as the key and iteratively masking off more and more low order bits of the address. This iterative lookup ensures that the **netmasks(4)** database can be used to specify the netmasks when variable length subnetmasks are used within a network number.

If a pseudo host name/pseudo network name is supplied as the netmask value, netmask data may be located in the **hosts** or **networks** database. Names are looked up by first using **gethostbyname(3NSL)**. If not found there, the names are looked up in **getnetbyname(3SOCKET)**. These interfaces may in turn use **nsswitch.conf(4)** to determine what data store(s) to use to fetch the actual value.

For both **inet** and **inet6**, the same information conveyed by *mask* can be specified as a *prefix_length* attached to the *address* parameter.

nud

Enables the neighbor unreachability detection mechanism on a point-to-point physical interface.

-nud

Disables the neighbor unreachability detection mechanism on a point-to-point physical interface.

plumb

Open the device associated with the physical interface name and set up the streams needed for **IP** to use the device. When used with a logical interface name, this command is used to create a specific named logical interface. An interface must be separately plumbed for use by IPv4 and IPv6. The *address_family* parameter controls whether the **ifconfig** command applies to IPv4 or IPv6.

Before an interface has been plumbed, the interface will not show up in the output of the **ifconfig -a** command.

private

Tells the **in.routed** routing daemon that a specified logical interface should not be advertised.

-private

Specify unadvertised interfaces.

removeif *address*

Remove the logical interface on the physical interface specified that matches the *address* specified. When the interface is part of a multipathing group, the logical interface will be removed from the physical interface in the group that holds the address.

router

Enable IP forwarding on the interface. When enabled, the interface is marked **ROUTER**, and IP packets can be forwarded to and from the interface.

-router

Disable IP forwarding on the interface. IP packets are not forwarded to and from the interface.

set

Set the *address*, *prefix_length* or both, for a logical interface.

standby

Marks the physical interface as a standby interface. If the interface is marked **STANDBY** and is part of the multipathing group, the interface will not be selected to send out packets unless some other interface in the group has failed and the network access has been failed over to this standby interface.

The status display shows "**STANDBY, INACTIVE**" indicating that that the interface is a standby

and is also inactive. **IFF_INACTIVE** will be cleared when some other interface belonging to the same multipathing group fails over to this interface. Once a failback happens, the status display will return to **INACTIVE**.

-standby

Turns off standby on this interface.

subnet

Set the subnet *address* for an interface.

tdst *tunnel_dest_address*

Set the destination address of a tunnel. The address should not be the same as the **dest_address** of the tunnel, because no packets leave the system over such a tunnel.

thoplimit *n*

Set the hop limit for a tunnel interface. The hop limit value is used as the **TTL** in the IPv4 header for the IPv6-in-IPv4 and IPv4-in-IPv4 tunnels. For IPv6-in-IPv6 and IPv4-in-IPv6 tunnels, the hop limit value is used as the hop limit in the IPv6 header.

token *address/prefix_length*

Set the IPv6 token of an interface to be used for address autoconfiguration.

```
example% ifconfig eri0 inet6 token ::1/64
```

trailers

This flag previously caused a nonstandard encapsulation of IPv4 packets on certain link levels. Drivers supplied with this release no longer use this flag. It is provided for compatibility, but is ignored.

-trailers

Disable the use of a "trailer" link level encapsulation.

tsrc *tunnel_src_address*

Set the source address of a tunnel. This is the source address on an outer encapsulating **IP** header. It must be an address of another interface already configured using **ifconfig**.

unplumb

Close the device associated with this physical interface name and any streams that **ifconfig** set up for IP to use the device. When used with a logical interface name, the logical interface is removed from the system. After this command is executed, the device name will no longer appear in the output of **ifconfig -a**.

up

Mark a logical interface "**up**". This happens automatically when assigning the first address to a logical interface. The **up** option enables an interface after an **ifconfig down**, which reinitializes the hardware.

usesrc [*name* | **none**]

Specify a physical interface to be used for source address selection. If the keyword **none** is used, then any previous selection is cleared.

When an application does not choose a non-zero source address using **bind(3SOCKET)**, the system will select an appropriate source address based on the outbound interface and the address selection rules (see **ipaddrsel(1M)**).

When **usesrc** is specified and the specified interface is selected in the forwarding table for output, the system looks first to the specified physical interface and its associated logical interfaces when selecting a source address. If no usable address is listed in the forwarding table, the ordinary selection rules apply. For example, if you enter:

```
# ifconfig eri0 usesrc vni0
```

...and **vni0** has address 10.0.0.1 assigned to it, the system will prefer 10.0.0.1 as the source address for any packets originated by local connections that are sent through **eri0**. Further examples are provided in the EXAMPLES section.

While you can specify any physical interface (or even loopback), be aware that you can also specify the virtual IP interface (see **vni(7D)**). The virtual IP interface is not associated with any physical hardware and is thus immune to hardware failures. You can specify any number of physical interfaces to use the source address hosted on a single virtual interface. This simplifies the configuration of routing-based multipathing. If one of the physical interfaces were to fail, communication would continue through one of the remaining, functioning physical interfaces. This scenario assumes that the reachability of the address hosted on the virtual interface is advertised in some manner, for example, through a routing protocol.

Because the **ifconfig preferred** option is applied to all interfaces, it is coarser-grained than the **usesrc** option. It will be overridden by **usesrc** and **setsrc** (route subcommand), in that order.

The use of the **usesrc** option is mutually exclusive of the IP multipathing **ifconfig** options, **group** and **standby**. That is, if an interface is already part of a IP multipathing group or specified as a **standby** interface, then it cannot be specified with a **usesrc** option, and vice-versa. For more details on IP multipathing, see **in.mpathd(1M)** and the .

xmit

Enable a logical interface to transmit packets. This is the default behavior when the logical interface is up.

-xmit

Disable transmission of packets on an interface. The interface will continue to receive packets.

zone *zonename*

Place the logical interface in zone *zonename*. The named zone must be active in the kernel in the ready or running state. The interface is unplumbed when the zone is halted or rebooted. The zone must be configured to be an shared-IP zone. **zonecfg(1M)** is used to assign network interface names to exclusive-IP zones.

-zone

Place IP interface in the global zone. This is the default.

OPERANDS

The *interface* operand, as well as address parameters that affect it, are described below.

interface

A string of one of the following forms:

- *name physical-unit*, for example, **eri0** or **ce1**
- *name physical-unit:logical-unit*, for example, **eri0:1**
- **ip.tunN** or **ip6.tunN**, for tunnels

If the interface name starts with a dash (-), it is interpreted as a set of options which specify a set of interfaces. In such a case, **-a** must be part of the options and any of the additional options below can be added in any order. If one of these interface names is given, the commands following it are applied to all of the interfaces that match.

-a

Apply the command to all interfaces of the specified address family. If no address family is supplied, either on the command line or by means of **/etc/default/inet_type**, then all address families will be selected.

-d

Apply the commands to all "down" interfaces in the system.

-D

Apply the commands to all interfaces not under **DHCP** (Dynamic Host Configuration Protocol) control.

-u

Apply the commands to all "up" interfaces in the system.

-Z

Apply the commands to all interfaces in the user's zone.

-4

Apply the commands to all IPv4 interfaces.

-6

Apply the commands to all IPv6 interfaces.

address_family

The address family is specified by the *address_family* parameter. The **ifconfig** command currently supports the following families: **inet** and **inet6**. If no address family is specified, the default is **inet**.

ifconfig honors the **DEFAULT_IP** setting in the `/etc/default/inet_type` file when it displays interface information. If **DEFAULT_IP** is set to **IP_VERSION4**, then **ifconfig** will omit information that relates to IPv6 interfaces. However, when you explicitly specify an address family (**inet** or **inet6**) on the **ifconfig** command line, the command line overrides the **DEFAULT_IP** settings.

address

For the IPv4 family (**inet**), the *address* is either a host name present in the host name data base (see **hosts(4)**) or in the Network Information Service (NIS) map **hosts**, or an IPv4 address expressed in the Internet standard "dot notation".

For the IPv6 family (**inet6**), the *address* is either a host name present in the host name data base (see **hosts(4)**) or in the Network Information Service (NIS) map **ipnode**, or an IPv6 address expressed in the Internet standard colon-separated hexadecimal format represented as `x:x:x:x:x:x:x:x` where *x* is a hexadecimal number between **0** and **FFFF**.

prefix_length

For the IPv4 and IPv6 families (**inet** and **inet6**), the *prefix_length* is a number between 0 and the number of bits in the address. For **inet**, the number of bits in the address is 32; for **inet6**, the number of bits in the address is 128. The *prefix_length* denotes the number of leading set bits in the netmask.

dest_address

If the *dest_address* parameter is supplied in addition to the *address* parameter, it specifies the address of the correspondent on the other end of a point-to-point link.

tunnel_dest_address

An address that is or will be reachable through an interface other than the tunnel being configured. This tells the tunnel where to send the tunneled packets. This address must not be the same as the interface destination address being configured.

tunnel_src_address

An address that is attached to an already configured interface that has been configured "up" with **ifconfig**.

INTERFACE FLAGS

The **ifconfig** command supports the following interface flags. The term "address" in this context refers to a logical interface, for example, **eri0:0**, while "interface " refers to the physical interface, for example, **eri0**.

ADDRCONF

The address is from stateless **addrconf**. The stateless mechanism allows a host to generate its own address using a combination of information advertised by routers and locally available information. Routers advertise prefixes that identify the subnet associated with the link, while the host generates an "interface identifier" that uniquely identifies an interface in a subnet. In the absence of information from routers, a host can generate link-local addresses. This flag is specific to IPv6.

ANYCAST

Indicates an **anycast** address. An **anycast** address identifies the nearest member of a group of systems that provides a particular type of service. An **anycast** address is assigned to a group of systems. Packets are delivered to the nearest group member identified by the **anycast** address instead of being delivered to all members of the group.

BROADCAST

This **broadcast** address is valid. This flag and **POINTTOPOINT** are mutually exclusive

CoS

This interface supports some form of Class of Service (CoS) marking. An example is the 802.1D user priority marking supported on **VLAN** interfaces.

DEPRECATED

This address is deprecated. This address will not be used as a source address for outbound packets unless there are no other addresses on this interface or an application has explicitly bound to this address. An IPv6 deprecated address will eventually be deleted when not used, whereas an IPv4 deprecated address is often used with IP network multipathing IPv4 test addresses, which are determined by the setting of the **NOFAILOVER** flag. Further, the **DEPRECATED** flag is part of the standard mechanism for renumbering in IPv6.

DHCP

DHCP is used to manage this address.

DUPLICATE

The logical interface has been disabled because the IP address configured on the interface is a duplicate. Some other node on the network is using this address. If the address was configured by DHCP or is temporary, the system will choose another automatically, if possible. Otherwise, the system will attempt to recover this address periodically and the interface will recover when the conflict has been removed from the network. Changing the address or netmask, or setting the logical interface to **up** will restart duplicate detection. Setting the interface to **down** terminates recovery and removes the **DUPLICATE** flag.

FAILED

The interface has failed. New addresses cannot be created on this interface. If this interface is part of an IP network multipathing group, a failover will occur to another interface in the group, if possible

FIXEDMTU

The MTU has been set using the **-mtu** option. This flag is read-only. Interfaces that have this flag set have a fixed MTU value that is unaffected by dynamic MTU changes that can occur when drivers notify IP of link MTU changes.

INACTIVE

Indicates that the interface is not currently being used for regular traffic by the system. New addresses cannot be created on this interface. The flag is set automatically on standby interfaces. It can also be set when the system detects that a failed interface has been repaired and **FAILBACK=no** is configured in **/etc/default/mpathd**. The flag is cleared when the interface fails or when a failover to that interface occurs.

LOOPBACK

Indicates that this is the loopback interface.

MIP

Indicates that mobile **IP** controls this interface.

MULTI_BCAST

Indicates that the broadcast address is used for multicast on this interface.

MULTICAST

The interface supports multicast. **IP** assumes that any interface that supports hardware broadcast, or

that is a point-to-point link, will support multicast.

NOARP

There is no address resolution protocol (**ARP**) for this interface that corresponds to all interfaces for a device without a broadcast address. This flag is specific to IPv4.

NOFAILOVER

This address will not failover if the interface fails. IP network multipathing test addresses must be marked **nofailover**.

NOLOCAL

The interface has no address, just an on-link subnet.

NONUD

NUD is disabled on this interface. **NUD** (neighbor unreachability detection) is used by a node to track the reachability state of its neighbors, to which the node actively sends packets, and to perform any recovery if a neighbor is detected to be unreachable. This flag is specific to IPv6.

NORTEXCH

The interface does not exchange routing information. For RIP-2, routing packets are not sent over this interface. Additionally, messages that appear to come over this interface receive no response. The subnet or address of this interface is not included in advertisements over other interfaces to other routers.

NOXMIT

Indicates that the address does not transmit packets. RIP-2 also does not advertise this address.

OFFLINE

Indicates that the interface has been offline. New addresses cannot be created on this interface. Interfaces in an IP network multipathing group are offline prior to removal and replacement using dynamic reconfiguration.

POINTOPOINT

Indicates that the address is a point-to-point link. This flag and **BROADCAST** are mutually exclusive

PREFERRED

This address is a preferred IPv6 source address. This address will be used as a source address for IPv6 communication with all IPv6 destinations, unless another address on the system is of more

appropriate scope. The **DEPRECATED** flag takes precedence over the **PREFERRED** flag.

PRIVATE

Indicates that this address is not advertised. For RIP-2, this interface is used to send advertisements. However, neither the subnet nor this address are included in advertisements to other routers.

ROUTER

Indicates that IP packets can be forwarded to and from the interface.

RUNNING

Indicates that the required resources for an interface are allocated. For some interfaces this also indicates that the link is up.

STANDBY

Indicates that this is a standby interface to be used on failures. Only interfaces in an IP network multipathing group should be designated as standby interfaces. If this interface is part of a IP network multipathing group, the interface will not be selected to send out packets unless some other interface in the group fails over to it.

TEMPORARY

Indicates that this is a temporary IPv6 address as defined in RFC 3041.

UNNUMBERED

This flag is set when the local IP address on the link matches the local address of some other link in the system

UP

Indicates that the interface is up, that is, all the routing entries and the like for this interface have been set up.

VIRTUAL

Indicates that the physical interface has no underlying hardware. It is not possible to transmit or receive packets through a virtual interface. These interfaces are useful for configuring local addresses that can be used on multiple interfaces. (See also the **-usesrc** option.)

XRESOLV

Indicates that the interface uses an IPv6 external resolver.

LOGICAL INTERFACES

Solaris **TCP/IP** allows multiple logical interfaces to be associated with a physical network interface. This allows a single machine to be assigned multiple **IP** addresses, even though it may have only one network interface. Physical network interfaces have names of the form *driver-name physical-unit-number*, while logical interfaces have names of the form *driver-name physical-unit-number:logical-unit-number*. A physical interface is configured into the system using the **plumb** command. For example:

```
example% ifconfig eri0 plumb
```

Once a physical interface has been "plumbed", logical interfaces associated with the physical interface can be configured by separate **-plumb** or **-addif** options to the **ifconfig** command.

```
example% ifconfig eri0:1 plumb
```

allocates a specific logical interface associated with the physical interface **eri0**. The command

```
example% ifconfig eri0 addif 192.168.200.1/24 up
```

allocates the next available logical unit number on the **eri0** physical interface and assigns an *address* and *prefix_length*.

A logical interface can be configured with parameters (*address*, *prefix_length*, and so on) different from the physical interface with which it is associated. Logical interfaces that are associated with the same physical interface can be given different parameters as well. Each logical interface must be associated with an existing and "up" physical interface. So, for example, the logical interface **eri0:1** can only be configured after the physical interface **eri0** has been plumbed.

To delete a logical interface, use the **-unplumb** or **-removeif** options. For example,

```
example% ifconfig eri0:1 down unplumb
```

will delete the logical interface **eri0:1**.

MULTIPATHING GROUPS

Physical interfaces that share the same IP broadcast domain can be collected into a multipathing group using the **group** keyword. Interfaces assigned to the same multipathing group are treated as equivalent and outgoing traffic is spread across the interfaces on a per-IP-destination basis. In addition, individual interfaces in a multipathing group are monitored for failures; the addresses associated with failed interfaces are automatically transferred to other functioning interfaces within the group.

For more details on IP multipathing, see **in.mpathd(1M)** and the . See **netstat(1M)** for per-IP-destination information.

CONFIGURING IPV6 INTERFACES

When an IPv6 physical interface is plumbed and configured "up" with **ifconfig**, it is automatically assigned an IPv6 link-local address for which the last 64 bits are calculated from the **MAC** address of the interface.

```
example% ifconfig eri0 inet6 plumb up
```

The following example shows that the link-local address has a prefix of **fe80::/10**.

```
example% ifconfig eri0 inet6
ce0: flags=2000841<UP,RUNNING,MULTICAST,IPv6>
      mtu 1500 index 2
      inet6 fe80::a00:20ff:fe8e:f3ad/10
```

Link-local addresses are only used for communication on the local subnet and are not visible to other subnets.

If an advertising IPv6 router exists on the link advertising prefixes, then the newly plumbed IPv6 interface will autoconfigure logical interface(s) depending on the prefix advertisements. For example, for the prefix advertisement **2001:0db8:3c4d:0:55::/64**, the autoconfigured interface will look like:

```
eri0:2: flags=2080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6>
      mtu 1500 index 2
      inet6 2001:0db8:3c4d:55:a00:20ff:fe8e:f3ad/64
```

Even if there are no prefix advertisements on the link, you can still assign global addresses manually, for example:

```
example% ifconfig eri0 inet6 addif \
2001:0db8:3c4d:55:a00:20ff:fe8e:f3ad/64 up
```

To configure boot-time defaults for the interface **eri0**, place the following entry in the **/etc/hostname6.eri0** file:

```
addif 2001:0db8:3c4d:55:a00:20ff:fe8e:f3ad/64 up
```

Configuring IPv6/IPv4 tunnels

An IPv6 over IPv4 tunnel interface can send and receive IPv6 packets encapsulated in an IPv4 packet. Create tunnels at both ends pointing to each other. IPv6 over IPv4 tunnels require the tunnel source and tunnel destination IPv4 and IPv6 addresses. Solaris 8 supports both automatic and configured tunnels. For automatic tunnels, an IPv4-compatible IPv6 address is used. The following demonstrates auto-tunnel configuration:

```
example% ifconfig ip.atun0 inet6 plumb
example% ifconfig ip.atun0 inet6 tsrc IPv4-address \
::IPv4 address/96 up
```

where **IPv4-address** is the IPv4 address of the interface through which the tunnel traffic will flow, and **IPv4-address, ::<IPv4-address>**, is the corresponding IPv4-compatible IPv6 address.

The following is an example of a configured tunnel:

```
example% ifconfig ip.tun0 inet6 plumb tsrc my-ipv4-address \
tdst peer-ipv4-address up
```

This creates a configured tunnel between **my-ipv4-address** and **peer-ipv4-address** with corresponding link-local addresses. For tunnels with global or site-local addresses, the logical tunnel interfaces need to be configured in the following form:

```
example% ifconfig ip.tun0 inet6 addif my-v6-address peer-v6-address up
```

For example,

```
example% ifconfig ip.tun0 inet6 plumb tsrc 109.146.85.57 \
tdst 109.146.85.212 up
example% ifconfig ip.tun0 inet6 addif 2::45 2::46 up
```

To show all IPv6 interfaces that are up and configured:

```
example% ifconfig -au6
ip.tun0: flags=2200851<UP,POINTOPOINT,RUNNING,MULTICAST,NUD,IPv6>
mtu 1480 index 3
inet tunnel src 109.146.85.57 tunnel dst 109.146.85.212
tunnel security settings --> use 'ipsecconf -ln -i ip.tun1'
tunnel hop limit 60
inet6 fe80::6d92:5539/10 --> fe80::6d92:55d4
ip.tun0:1: flags=2200851<UP,POINTOPOINT,RUNNING,MULTICAST,NUD,IPv6>
mtu 1480 index 3
inet6 2::45/128 --> 2::46
```

In the output above, note the line that begins with "**tunnel security settings**". The content of this line varies according to whether and how you have set your security settings. See "Display of Tunnel Security Settings," below.

Configuring IPv4/IPv6 Tunnels

An IPv4 over IPv6 tunnel interface can send and receive IPv4 packets encapsulated in an IPv6 packet. Create tunnels at both ends pointing to each other. IPv4 over IPv6 tunnels require the tunnel source and tunnel destination IPv6 and IPv4 addresses. The following demonstrates auto-tunnel configuration:

```
example% ifconfig ip6.tun0 inet plumb tsrc my-ipv6-address \  
          tdst peer-ipv6-address my-ipv4-address \  
          peer-ipv4-address up
```

This creates a configured tunnel between *my-ipv6-address* and *peer-ipv6-address* with *my-ipv4-address* and *peer-ipv4-address* as the endpoints of the point-to-point interface, for example:

```
example% ifconfig ip6.tun0 inet plumb tsrc fe80::1 tdst fe80::2 \  
          10.0.0.208 10.0.0.210 up
```

To show all IPv4 interfaces that are up and configured:

```
example% ifconfig -au4  
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1  
    inet 127.0.0.1 netmask ff000000  
eri0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu 1500 \  
index 2  
    inet 172.17.128.208 netmask ffffff00 broadcast 172.17.128.255  
ip6.tun0: flags=10008d1<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST,IPv4> \  
mtu 1460  
index 3  
    inet6 tunnel src fe80::1 tunnel dst fe80::2  
    tunnel security settings --> use 'ipseconf -ln -i ip.tun1'  
    tunnel hop limit 60 tunnel encapsulation limit 4  
    inet 10.0.0.208 --> 10.0.0.210 netmask ff000000
```

In the output above, note the line that begins with "**tunnel security settings**". The content of this line varies according to whether and how you have set your security settings. See "Display of Tunnel Security Settings," below.

Display of Tunnel Security Settings

The **ifconfig** output for tunneled interfaces indicates security settings, if present, for a tunnel. The content of the line showing your settings differs depending on how you have made your settings:

- If you set your security policy using the **ifconfig -auth_algs**, **-encr_algs**, and **-encr_auth_algs** options and do not use **ipseconf(1M)**, **ifconfig** displays your settings for each of these options.
- If you set your security policy using **ipseconf(1M)** with the **tunnel** keyword (the preferred method), **ifconfig** displays:

```
tunnel security settings --> use 'ipseconf -ln -i ip.tun1'
```

...in effect, hiding your settings from those without privileges to view them.

If you do not set security policy, using either **ifconfig** or **ipseccnf**, there is no tunnel security setting displayed.

EXAMPLES

Example 1 Using the **ifconfig** Command

If your workstation is not attached to an Ethernet, the network interface, for example, **eri0**, should be marked "down" as follows:

```
example% ifconfig eri0 down
```

Example 2 Printing Addressing Information

To print out the addressing information for each interface, use the following command:

```
example% ifconfig -a
```

Example 3 Resetting the Broadcast Address

To reset each interface's broadcast address after the netmasks have been correctly set, use the next command:

```
example% ifconfig -a broadcast +
```

Example 4 Changing the Ethernet Address

To change the Ethernet address for interface **ce0**, use the following command:

```
example% ifconfig ce0 ether aa:1:2:3:4:5
```

Example 5 Configuring an IP-in-IP Tunnel

To configure an IP-in-IP tunnel, first plumb it with the following command:

```
example% ifconfig ip.tun0 plumb
```

Then configure it as a point-to-point interface, supplying the tunnel source and the tunnel destination:

```
example% ifconfig ip.tun0 myaddr mydestaddr tsrc another_myaddr \
        tdst a_dest_addr up
```

Use **ipsecconf**(1M), as described above, to configure tunnel security properties.

Example 6 Configuring 6to4 Tunnels

To configure 6to4 tunnels, use the following commands:

```
example% ifconfig ip.6to4tun0 inet6 plumb
example% ifconfig ip.6to4tun0 inet6 tsrc IPv4-address 6to4-address/64 up
```

IPv4-address denotes the address of the encapsulating interface. *6to4-address* denotes the address of the local IPv6 address of form *2002:IPv4-address:SUBNET-ID:HOSTID*.

The long form should be used to resolve any potential conflicts that might arise if the system administrator utilizes an addressing plan where the values for **SUBNET-ID** or **HOSTID** are reserved for something else.

After the interface is plumbed, a 6to4 tunnel can be configured as follows:

```
example% ifconfig ip.6to4tun0 inet6 tsrc IPv4-address up
```

This short form sets the address. It uses the convention:

```
2002:IPv4-address::1
```

The **SUBNET-ID** is 0, and the **HOSTID** is 1.

Example 7 Configuring IP Forwarding on an Interface

To enable IP forwarding on a single interface, use the following command:

```
example% ifconfig eri0 router
```

To disable IP forwarding on a single interface, use the following command:

```
example% ifconfig eri0 -router
```

Example 8 Configuring Source Address Selection Using a Virtual Interface

The following command configures source address selection such that every packet that is locally generated with no bound source address and going out on **qfe2** prefers a source address hosted on **vni0**.

```
example% ifconfig qfe2 usesrc vni0
```

The **ifconfig -a** output for the **qfe2** and **vni0** interfaces displays as follows:

```
qfe2: flags=1100843<UP,BROADCAST,RUNNING,MULTICAST,ROUTER,IPv4> mtu
1500 index 4
usesrc vni0
inet 1.2.3.4 netmask fffffff0 broadcast 1.2.3.255
ether 0:3:ba:17:4b:e1
vni0: flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL>
mtu 0 index 5
srcof qfe2
inet 3.4.5.6 netmask ffffffff
```

Observe, above, the **usesrc** and **srcof** keywords in the **ifconfig** output. These keywords also appear on the logical instances of the physical interface, even though this is a per-physical interface parameter. There is no **srcof** keyword in **ifconfig** for configuring interfaces. This information is determined automatically from the set of interfaces that have **usesrc** set on them.

The following command, using the **none** keyword, undoes the effect of the preceding **ifconfig usesrc** command.

```
example% ifconfig qfe2 usesrc none
```

Following this command, **ifconfig -a** output displays as follows:

```

qfe2: flags=1100843<UP,BROADCAST,RUNNING,MULTICAST,ROUTER,IPv4> mtu
1500 index 4
inet 1.2.3.4 netmask ffffff00 broadcast 1.2.3.255
ether 0:3:ba:17:4b:e1
vni0: flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL>
mtu 0 index 5
inet 3.4.5.6 netmask ffffffff

```

Note the absence of the **usesrc** and **srcif** keywords in the output above.

Example 9 Configuring Source Address Selection for an IPv6 Address

The following command configures source address selection for an IPv6 address, selecting a source address hosted on **vni0**.

```
example% ifconfig qfe1 inet6 usesrc vni0
```

Following this command, **ifconfig -a** output displays as follows:

```

qfe1: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 3
usesrc vni0
inet6 fe80::203:baff:fe17:4be0/10
ether 0:3:ba:17:4b:e0
vni0: flags=2002210041<UP,RUNNING,NOXMIT,NONUD,IPv6,VIRTUAL> mtu 0
index 5
srcif qfe1
inet6 fe80::203:baff:fe17:4444/128
vni0:1: flags=2002210040<RUNNING,NOXMIT,NONUD,IPv6,VIRTUAL> mtu 0
index 5
srcif qfe1
inet6 fec0::203:baff:fe17:4444/128
vni0:2: flags=2002210040<RUNNING,NOXMIT,NONUD,IPv6,VIRTUAL> mtu 0
index 5
srcif qfe1
inet6 2000::203:baff:fe17:4444/128

```

Depending on the scope of the destination of the packet going out on **qfe1**, the appropriately scoped source address is selected from **vni0** and its aliases.

Example 10 Using Source Address Selection with Shared-IP Zones

The following is an example of how the **usesrc** feature can be used with the **zones(5)** facility in Solaris. The following commands are invoked in the global zone:

```
example% ifconfig hme0 usesrc vni0
example% ifconfig eri0 usesrc vni0
example% ifconfig qfe0 usesrc vni0
```

Following the preceding commands, the **ifconfig -a** output for the virtual interfaces would display as:

```
vni0: flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL>
  mtu 0 index 23
  srcof hme0 eri0 qfe0
  inet 10.0.0.1 netmask ffffffff
vni0:1:
  flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL> mtu 0
  index 23
  zone test1
  srcof hme0 eri0 qfe0
  inet 10.0.0.2 netmask ffffffff
vni0:2:
  flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL> mtu 0
  index 23
  zone test2
  srcof hme0 eri0 qfe0
  inet 10.0.0.3 netmask ffffffff
vni0:3:
  flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL> mtu 0
  index 23
  zone test3
  srcof hme0 eri0 qfe0
  inet 10.0.0.4 netmask ffffffff
```

There is one virtual interface alias per zone (**test1**, **test2**, and **test3**). A source address from the virtual interface alias in the same zone is selected. The virtual interface aliases were created using **zonecfg(1M)** as follows:

```
example% zonecfg -z test1
zonecfg:test1> add net
zonecfg:test1:net> set physical=vni0
zonecfg:test1:net> set address=10.0.0.2
```

The **test2** and **test3** zone interfaces and addresses are created in the same way.

Example 11 Turning Off DHCPv6

The following example shows how to disable automatic use of DHCPv6 on all interfaces, and immediately shut down DHCPv6 on the interface named **hme0**. See **in.ndpd(1M)** and **ndpd.conf(4)** for more

information on the automatic DHCPv6 configuration mechanism.

```
example% echo ifdefault StatefulAddrConf false >> /etc/inet/ndpd.conf
example% pkill -HUP -x in.ndpd
example% ifconfig hme0 dhcp release
```

FILES

/etc/netmasks

Netmask data.

/etc/default/inet_type

Default Internet protocol type.

ATTRIBUTES

See **attributes(5)** for descriptions of the following attributes:

```
tab() box; cw(2.73i) |cw(2.77i) lw(2.73i) |lw(2.77i) ATTRIBUTE TYPEATTRIBUTE VALUE _
AvailabilitySUNWcsu _ T{ Interface Stability for command-line options T}Committed _ Interface Sta-
bility for command outputUncommitted
```

SEE ALSO

dhcpinfo(1), **dhcpcagent(1M)**, **in.mpathd(1M)**, **in.ndpd(1M)**, **in.routed(1M)**, **ipsecconf(1M)**, **ndd(1M)**, **netstat(1M)**, **zoneadm(1M)**, **zonecfg(1M)**, **ethers(3SOCKET)**, **gethostbyname(3NSL)**, **getnetbyname(3SOCKET)**, **hosts(4)**, **inet_type(4)**, **ndpd.conf(4)**, **netmasks(4)**, **networks(4)**, **nsswitch.conf(4)**, **attributes(5)**, **privileges(5)**, **zones(5)**, **arp(7P)**, **ipsecah(7P)**, **ipsecesp(7P)**, **tun(7M)**

DIAGNOSTICS

ifconfig sends messages that indicate if:

- the specified interface does not exist
- the requested address is unknown
- the user is not privileged and tried to alter an interface's configuration

NOTES

Do not select the names **broadcast**, **down**, **private**, **trailers**, **up** or other possible option names when you choose host names. If you choose any one of these names as host names, it can cause unusual problems that are extremely difficult to diagnose.