Native IPv6

If you already have native connectivity, Debian should pick it up automatically from router advertisments, and generate addresses for you. You can also use the "static" method in /etc/network/interfaces to set up addresses manually. PPP users may have to add +ipv6 to /etc/ppp/options, as not all PPP RAS servers advertise IPv6 support, even if they will negotiate it upon request.

If your router doesn't have advertisements, you may want to try to bring ipv6 manually:

```
Â ip -6 addr add 2002:c000:203::1/64 dev eth0
```

Of course, replace the address by the one corresponding to your IPv4 or anything that you want to assign to your eth0. You can remove an IPv6 from the device the same way:

```
Â ip -6 addr del 2002:c000:203::1/64 dev eth0
```

If you have both native IPv6 and IPv4 (a dual-stack host), you may still want to use 6to4 addresses as well as your native addresses; IPv6 specifies address selection rules that ensure that native hosts will use your native IPv6 connectivity, but 6to4 hosts will use your 6to4 connectivity (removing public relay routers, and reducing the number of points of failure in your setup).

Â

IPv6 6to4 Configuration

You may not know this, but if you have a public IPv4 address you already have IPv6 addresses reserved. There is a transitional system called *6to4* (sometimes written stf or 6 to 4) that maps any IPv4 address an entire subnet of IPv6 addresses. You can use it to assign a single address to your system, or as the prefix for your local IPv6 network. You can also use the method for making a private IPv6 network by using a private IPv4 (
RFC 1918
) address instead of the deprecated
"site-local"
fec0::/10 addresses.

To configure 6to4, you need to calculate an IPv6 address, based on your existing IPv4 address. You can do so using the shell. In this example the IPv4 address is 192.0.2.3, you'll replace that with your actual public internet address. The ::1 on the end says this is machine address 1 in subnet 0. You have 16 bits of network and 64 bits of host addresses available.

```
Â $ printf "2002:%02x%02x:%02x%02x::1n" 192 0 2 3  Â 2002:c000:0203::1
```

or use ipv6calc which gives you the prefix

Â

```
  $ ipv6calc --quiet --action conv6to4 192.0.2.3  2002:c000:203::
```

Â

Now edit /etc/network/interfaces, and add a stanza like this using a dedicated tunnel device named tun6to4 (or whatever you wish to call it):

Â

```
  auto tun6to4  iface tun6to4 inet6 v4tunnel         address 2002:c000:0203::1         netmask 16
      gateway ::192.88.99.1         endpoint any         local 192.0.2.3 #fits address
```

Â

Or using the deprecated sit0 interface:

Â

```
  auto sit0  iface sit0 inet6 static         address 2002:c000:0203::1         netmask 16
gateway ::192.88.99.1
```

Â

The 192.88.99.1 address is a special anycast address that points to the nearest 6to4 router. Assuming that your nearest 6to4 router work, all you should need to do is ifup tun6to4 (or sit0) and enjoy your new IPv6 connectivity!

Experienced IPv6 users will wonder why the netmask for sit0 is /16, not /48; by setting the netmask to /16, you instruct your system to send packets directly to the IPv4 address of other 6to4 users; if it was /48, you'd send packets via the nearest relay router, increasing latency.

The sit0 example may result in a responses with an IPv4 address that doesn't match the 6to4 address on a multihomed host. Also the generic tunnel device sit0 doesn't let you specify filtering per device.

See also:

- http://tldp.org/HOWTO/Linux+IPv6-HOWTO/configuring-ipv6to4-tunnels.html

- 6to4

Â

Setting up a 6to4 relay router using Debian

Â

If you have a dual-stacked host (one with both IPv4 and IPv6 native connectivity), you may wish to offer a relay router for the rest of your network. A relay router also helps with connectivity for IPv6-only hosts on your network; rather than relying on a public 6to4 relay router to communicate with 6to4 hosts, your router can encapsulate IPv6 into IPv4.

There are two ways to do this; one is to have the relay on a normal address, which you share with your clients. The other is to add 192.88.99.1/24 to your IPv4 addresses and routing, so that clients that use 6to4 will pick up your relay router automatically. If you run a relay router on an address other than 192.88.99.1, you will need to change the gateway on your clients to match.

A relay router should already be configured to forward IPv6 packets natively for its clients; you then add a sit0 interface **without** a gateway for the 6to4 relay. In /etc/network/interfaces, this looks like:

Â

```
  auto sit0  iface sit0 inet6 static         address 2002:c000:0203::1         netmask 16
```

Â

Note that this is almost identical to a plain 6to4 setup; the only difference is that as you already have native connectivity, you do not need to use a relay router to act as your gateway.

Â

IPv6 Tunnel Configuration

Â

An IPv6 tunnel can be configured using /etc/network/interfaces on Debian:

Â

```
  auto 6in4  iface 6in4 inet6 v4tunnel    address [Your IPv6 Endpoint]    netmask [Prefix Length]
 endpoint [PoP IPv4 Endpoint]    gateway [PoP IPv6 Endpoint]    ttl 64    up ip link set mtu 1280
dev $IFACE
```

Â

Now you can use ifup(8) and ifdown(8) to control your tunnel. What's even better about it, is that the tunnel is set up at boot time, because in /etc/init.d/networking, "ifup -a" is called, which brings up all interfaces from /etc/network/interfaces configured with the "auto" option.

Â

Automated Tunnel Configuration

Â

Try the aiccu , for SixXS or 'tspc', for Hexago package to automate most of the tunnel configuration.

Â

Manual Tunnel Configuration

Â

Hurricane Electric operates a free tunnel broker , with simple web based registration and forums. Examples for Debian using ifconfig or ip commands available.
Configuration instructions for use with ifupdown (/etc/network/interfaces)
are available on the tunnelbroker.net forum.

Â

How to turn off IPv6

Â

Â

In squeeze

Â

-

Disable ipv6 in kernel : *echo net.ipv6.conf.all.disable_ipv6=1 > /etc/sysctl.d/disableipv6.conf* will disable ipv6 at next reboot.

- fetchmail will stop sending dns AAAA queries.
-

If you've built a custom kernel with IPv6 as a module be aware that due to a race condition with the init scripts you'll need to load the ipv6 module before the procps init script is run (see /usr/share/doc/procps/README.Debian and 507788)

- You will probably need to comment-out any IPv6 address in /etc/hosts (especially the one for loopback) otherwise ssh will fail to forward ports (or you must always use -4 to ssh).

- In exim4:
-

put *disable_ipv6 = true* into your exim configuration file

-

run *update-exim4.conf*

- then restart exim4

- In sshd:
-

put *AddressFamily* *inet* into /etc/ssh/sshd_config

- restart sshd: /etc/init.d/ssh restart

-

Change /etc/avahi/avahi-daemon.conf to say *use-ipv6=no*

Swisscom DSL with the latest netopia router will not work with Lenny out of the box. For example ssh only works the *-4* option despite

-

taking all steps recommended in http://wiki.debianforum.de/IPv6Deaktivieren

-

in /etc/modprobe.d/blacklist.conf setting *blacklist ipv6*