

Configuring IPv6 on Solaris 8

This paper will describe in detail how to configure a Sun Solaris 8 machine to correctly communicate over IPv6.

Native6Group, based in Seattle Washington, is a private partnership focused on helping enterprises transition their IPv4-based legacy networks to IPv6 – the emerging global IP platform. Our mission is to give our customers first-mover advantages in the early adoption of IPv6, and ensure that enterprises of all kinds plan their future networks starting now – allowing them to transition safely and securely with a minimum of business risk. We offer transition and network consulting, IPv6 training and education, and IPv6 device testing. To be successful, a business must plan for the rapid adoption of IPv6 starting today. For more information, please visit Native6Group's website at www.native6group.com.



www.native6group.com

Robert C Zilbauer Jr is a Native6Group contributor and a Sun Solaris Guru – among other talents. He has 11 years' experience with Sun operating systems and has recently been thrown headlong into the realm of IPv6. Robert can be reached through Native6Group.

Gathering the Pieces

Here's a brief list of what you'll need for this project:

- A platform capable of supporting Solaris 8. A Sun Sparc based system would work quite well. Also, most modern Intel or Intel compatible systems will fit the bill. I'd recommend at least 1.5gig of disk space (for a comfortable OS install plus any 3rd party niceties you may want) and at least 128meg of RAM. Of course, in the case of disk space and RAM, more is always better.
- Network information such as gateway address, netmask, and the IPv6 address assigned to your new system (if a static address is desired), etc.
- The Sun Solaris 8 software itself. This software is available for download from Sun Microsystems for free (<http://www.sun.com/software/solaris/binaries/get.html>) or can be purchased for a media fee of \$75 (as of this writing).
- Prerequisites: Zama Doc #1000 ("Configuring IPv4 on Solaris 8")

Static or Automatic?

At this point, we'll assume you have Solaris 8 installed, you chose to enable the IPv6 stack during the OS installation, and your IPv4 network connectivity is configured and working. If any of this is not true, you may want to refer to Zama Doc #1000 ("Configuring IPv4 on Solaris 8") for a detailed walk through. Otherwise, you're ready to continue.

The first thing you'll need to decide is how you want your IPv6 address defined. You have two choices. You can either have it automatically configured via the Neighbor Discovery protocol or you can define your IPv6 address statically. We'll describe the automatic configuration method first since that's the easiest.

Obtaining the IP through Auto-Configuration

On the Solaris side of things, this is the easiest way to configure your IPv6 address. Of course, this presupposes that there's a router on your network running the Neighbor Discovery protocol and advertising the correct IPv6 address prefix. Assuming you have such a beast available, automatic configuration of your IP address is as simple as using the *touch* command.

You'll need to know the name of the interface over which you want to use IPv6. The primary interface on a Sun machine is often `hme0`. On an Intel based machine it is often `e1x10`. However, if you don't know, you can usually use the *ifconfig -a* command to check. The output of that command will look something like this:

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.168.25.5 netmask ffffffff00 broadcast 192.168.25.255
znb0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.168.0.2 netmask ffffffff00 broadcast 192.168.0.255
```

The above example is from a Sun machine with two ethernet interfaces. The first entry, `lo0`, is the loopback interface. The second entry is the primary interface, `hme0`, and the third entry is an additional interface, `znb0`. In our case, we'll be using the primary interface for our IPv6 traffic, but the same steps would apply to any other interface on the machine. You'd just use the name of your chosen interface where ever `hme0` is referenced in this paper.

Now that we know the name of the interface we want to use, setting up auto configuration of your IPv6 address is simple. Just create an empty file called `/etc/hostname6.<interface name>` and you're done. The easiest way to do this is by using the `touch` command. In our case, we'd do the following:

```
flotsam# touch /etc/hostname6.hme0
```

Then, reboot. The "neighbor discovery protocol daemon" (see the `in.ndpd(1m)` man page for more information) will get the IPv6 prefix from your router, tack on your machine's 64-bit Extended Unique Identifier (EUI-64) address, and set the resulting address as the IPv6 address for the interface you specified.

Defining a Static IPv6 Address

If a static address is what you're after, don't go anywhere just yet. You've still got some work to do. As with automatic configuration, you still need an `/etc/hostname6.<interface>` file. However, when using a static address, this file is not empty.

In keeping with our example (using `hme0` as our IPv6 interface), you'd need your `/etc/hostname6.hme0` to contain one line with the following format:

```
addif host.name.domain/mask up
```

In our case, we want our new machine, `flotsam.mydomain.com`, to come up with the IPv6 address `3ffe:80f0:1:3:a00:20ff::5`. To do this we would edit `/etc/hostname6.hme0` and give it the following contents:

```
addif flotsam.mydomain.com/64 up
```

Then, in `/etc/inet/ipnodes`, we would add an entry for the static IP we want our machine to be known by:

```
3ffe:80f0:1:3:a00:20ff::5          flotsam.mydomain.com          flotsam
```

[Note: If you're familiar with configuring an IPv4 interface, you'll notice that the relationship between `/etc/hostname6.hme0` and `/etc/inet/ipnodes` is very similar to their IPv4 counterparts: `/etc/hostname.hme0` and `/etc/hosts`.]

Where your IPv6 default route is concerned, you should be fine with the auto-configured value. Even though you have your IP address defined statically, the Neighbor Discovery process should retrieve a valid IPv6 default route from your properly configured router.

Finally, reboot the machine. When it comes up, you should have both your IPv4 and your IPv6 interfaces configured. When you take a look at your interface configuration using `ifconfig -a` you should see something similar to the following:

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.168.25.5 netmask ffffffff broadcast 192.168.25.255
    ether 0:1:2:c4:d:ee
lo0: flags=2000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
    inet6 ::1/128
hme0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
    ether 0:1:2:c4:d:ee
    inet6 fe80::201:2ff:fec4:dee/10
hme0:1: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
```

```
inet6 3ffe:80f0:1:3:a00:20ff::5/64
```

In addition, you should use the `netstat -rn` command to verify that your routing tables are configured correctly. They should look something like this:

```
flotsam# netstat -rn
```

```
Routing Table: IPv4
```

Destination	Gateway	Flags	Ref	Use	Interface
192.168.25.0	192.168.25.5	U	1	172	hme0
224.0.0.0	192.168.25.5	U	1	0	hme0
default	192.168.25.1	UG	1	545	
127.0.0.1	127.0.0.1	UH	3	12	lo0

```
Routing Table: IPv6
```

Destination/Mask	Gateway	Flags	Ref	Use	If
3ffe:80f0:1:3::/64	3ffe:80f0:1:3:a00:20ff::5	U	1	0	elx10:1
fe80::/10	fe80::a00:20ff:fec5:6fa0	U	1	0	hme0
ff00::/8	fe80::a00:20ff:fec5:6fa0	U	1	0	hme0
default	fe80::a00:20ff:fec5:6fa0	U	1	0	hme0
::1	::1	UH	1	0	lo0

Now, make sure your IPv6 connectivity is working. Ideally, you have another machine on your network configured for IPv6 to test against. Try to *ping* that other machine's IPv6 address (assuming that you know it will return a ping). If that works, then you've got IPv6 properly configured for your interface.

DNS Configuration

One more step to take is to configure your new machine to use DNS to look up IPv6 hosts on the network. This step is not required for network connectivity, but it does make life easier. You'll need to edit your `/etc/nsswitch.conf` file to do this.

Just like with IPv4 and the `hosts` entry, you should add `dns` to the `ipnodes` entry to allow DNS lookups for IPv6 addresses. After making that change, your `hosts` and `ipnodes` lines in `/etc/nsswitch.conf` will look like this:

```
hosts:      files dns
ipnodes:    files dns
```

And there you have it. You've just completed the configuration for an IPv6 interface on a Solaris 8 machine. Well done!

References

Solaris Intel Spotlight

<http://www.sun.com/software/intel/>

"Implementing an IPv6 IPF Firewall"

John Spence & Grant Furness