



OPENSOLARIS NETWORK ADMINISTRATION For Administrators Familiar with Linux

> **OpenSolaris™** How To Guides



Jeff McMeekin, OpenSolaris Networking Product Manager

Version 1.1 | Last updated: 09/14/09

About This OpenSolaris How To Guide

This document is for people who are familiar with command line configuring and monitoring network connections on Linux systems and who want to do the same on OpenSolaris systems. While there are differences among the many Linux distributions, most Linux administrators are probably familiar with the Linux commands and configuration files referenced in this document.

Contents

Overview	Page 1
Interface Names	Page 1
Persistent Configuration Storage	Page 1
pfexec	Page 2
Plumbing and ifconfig	Page 2
Enabling Manual Network Configuration	Page 3
Listing Network Interfaces	Page 3
Renaming Network Interfaces	Page 3
Configuring an Interface	Page 4
Configuring a Route	Page 4
dladm	Page 5
nsswitch	Page 5
Resolv.conf	Page 6
Tcpwrappers	Page 6
Ipfilters and Iptables	Page 7
Network Debugging Tools	Page 7
General OpenSolaris System Administration Notes	Page 7
Root: User (Login) versus Role	Page 7
Man Pages	Page 7
IPS: Image Packaging System	Page 8
SMF: Service Management Framework	Page 9
Miscellaneous System Configuration and Status	Page 10
Appendix A: Ethtool Equivalent Commands	Page 11

OpenSolaris Networking How To Guide

Overview

On Linux immediate changes to network configuration are made with commands like these:

```
ifconfig
ip
route
ethtool
```

Permanent changes that take effect on reboot or on restart of the network service are made in config files like these:

```
/etc/sysconfig/network-scripts/ifcfg-<interface>
/etc/sysconfig/network-scripts/route-<interface>
```

On OpenSolaris the default behavior is to configure network interfaces automatically. If automatic configuration is disabled (see [Enabling Manual Network Configuration](#) below) then immediate changes to network configuration are made with these commands:

```
ifconfig
route
dladm
```

Persistent changes may be made in files or via commands. For example, these files are similar to the Linux `ifcfg-<interface>` files:

```
/etc/hostname.<interface>
```

while the `route` command has an option for making route changes persistent:

```
route -p ...
```

Interface Names

On Linux physical interface names have the form `eth<n>`, for example `eth0`. On OpenSolaris the names are descriptive of the interface driver (see [Listing Network Interfaces](#) below). Examples of interface names:

```
pcn0
bge0
e1000g0
```

On OpenSolaris interfaces can be renamed (see [Renaming Network Interfaces](#) below).

Persistent Configuration Storage

On Linux the configuration for an interface is stored in `ifcfg-<interface>`. On OpenSolaris that information is arranged a little differently.

IP address and MAC: The IP address is stored in `/etc/hostname.<interface>`. Host names are allowed in place of IP addresses, but that may not be a good idea if DNS or NIS is being used to look up host names (see [nsswitch](#) below); best practice is to use IP addresses in the `/etc/hostname.<interface>` config files. The MAC (usually an ethernet address) is optional.

Examples:

```
echo 10.10.10.17 > /etc/hostname.bge0
echo 10.10.10.17/24 > /etc/hostname.bge0
echo 10.10.10.17 ether 11:22:33:aa:bb:ff > /etc/hostname.bge0
echo host.my.domain > /etc/hostname.bge0
```

Linux does not guarantee that a particular port will always come up as the same interface (eth0, etc.) after a reboot, so it is common practice on Linux to specify the MAC along with the network configuration. OpenSolaris keeps a persistent mapping of physical devices to device names, so interface names stay the same *even if hardware is added or removed* unless the system is reconfigured (`reboot -r` or `sys-unconfig`), so it is not necessary to specify the MAC in `hostname.<interface>`, although it is allowed as in the example above.

Netmask: While it is possible to specify a netmask in `/etc/hostname.<interface>` (see the example above), the netmask usually comes from the netmask database as specified in the `nsswitch` settings (see `nsswitch` below). Example:

```
% grep netmask /etc/nsswitch.conf
netmasks: files
% grep 10.10.17.80 /etc/netmasks
10.10.17.80      255.255.255.240
% cat /etc/hostname.e1000g0
10.10.17.93
% ifconfig e1000g0
e1000g0:
flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu
1500 index 2
      inet 10.10.17.93 netmask ffffffff broadcast 10.10.17.95
```

Gateways: On Linux, gateways may be listed in `/etc/sysconfig/network` or in the `ifcfg-<interface>` files. On OpenSolaris gateways are listed in `/etc/defaultrouter`.

Static Routes: On Linux static routes are stored on a per-interface basis in `route-<interface>`. On OpenSolaris static routes are stored in `/etc/inet/static_routes`, but this file should not be edited by hand; rather, this file is maintained by the '-p' option to the route command.

pfexec

In the following sections many commands are run via `pfexec` or in a shell obtained via `pfexec`. The `pfexec` command does for Solaris roles what the `sudo` command does for normal accounts. See **Root: User (Login) versus Role** below for details.

Plumbing and ifconfig

On OpenSolaris an interface has to be "plumbed" before it can be configured, and unplumbed interfaces do not show up in `ifconfig -a`. Plumbing an interface sets up the software in the kernel to connect the interface to the TCP/IP stack. An interface that has a persistent configuration is automatically plumbed.

Plumb an interface:

```
% pfexec bash
# ifconfig e1000g0 plumb
```

The `ifconfig` command differs between Linux and OpenSolaris. The following illustrates two simple but notable differences:

	Linux	OpenSolaris
<code>ifconfig</code>	show all active interfaces	long 'usage' message
<code>ifconfig <interface></code>	show <interface>	show <interface>
<code>ifconfig -a</code>	show all interfaces	show all plumbed interfaces

To see all interfaces on OpenSolaris use `dladm show-link` (see [Listing Network Interfaces](#) below). Alternatively, plumb all interfaces:

```
% pfexec ifconfig -a plumb
% ifconfig -a
%
```

Enabling Manual Network Configuration

On many networks with DHCP servers it is simplest to let the Network AutoMagic (NWAM) daemon configure the network. To take full control of the network configuration, disable automatic configuration and enable manual configuration:

```
% pfexec bash
# svcadm disable network/physical:nwam
# svcadm enable network/physical:default
```

It is best to do this on the system console rather than over a network connection, since the network connection may be lost during reconfiguration.

Listing Network Interfaces

List all interfaces whether or not they are plumbed:

```
% dladm show-link
LINK          CLASS      MTU      STATE    OVER
e1000g0      phys      1500    up       --
%
```

Show the state of the interfaces (similar to Linux `ethtool`):

```
% dladm show-phys e1000g0
LINK          MEDIA          STATE    SPEED  DUPLEX  DEVICE
e1000g0      Ethernet      up       1000   full   bge0
% dladm show-dev e1000g0
LINK          STATE    SPEED  DUPLEX
e1000g       up       1000Mb full
```

Renaming Network Interfaces

Network interfaces can be renamed, for example:

```
# dladm show-link
LINK          CLASS      MTU      STATE    OVER
e1000g0      phys      1500    up       --
# dladm rename-link e1000g0 eth0
# dladm show-link
LINK          CLASS      MTU      STATE    OVER
eth0         phys      1500    unknown --
#
```

Configuring an Interface

Plumb if necessary and add an IP address:

```
% pfexec bash
# ifconfig e1000g0 plumb
# ifconfig e1000g0 10.10.17.93/28 broadcast + up

    note: "/28" in the above refers to a net mask of 0xfffff0 or 255.255.255.240

# ifconfig e1000g0
e1000g0:
flags=201000863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index
2
    inet 10.10.17.93 netmask fffffff0 broadcast 10.10.17.95
    ether 0:c:29:ac:9:5f
```

Make a persistent configuration:

```
% pfexec bash
# echo 10.10.17.93 > /etc/hostname.e1000g0
# svcadm restart network/physical:default
```

Configuring a Route

The route command syntax differs among versions of Unix and Linux, so be sure to check the man page, but adding and deleting routes is fairly straightforward:

```
% pfexec bash
# netstat -nr | grep 172
# route add net 172.16.17.0/24 10.10.17.81
add net 172.16.17.0/24: gateway 10.10.17.81
# netstat -nr | grep 172
172.16.17.0          10.10.17.81          UG          1          0
# route delete net 172.16.17.0/24 10.10.17.81
delete net 172.16.17.0/24: gateway 10.10.17.81
# netstat -nr | grep 172
```

Use the -p option to make persistent route changes:

```
% pfexec bash
# route -p show
No persistent routes are defined
# route -p add net 172.16.17.0/24 10.10.17.81
add net 172.16.17.0/24: gateway 10.10.17.81
add persistent net 172.16.17.0/24: gateway 10.10.17.81
# route -p show
persistent: route add net 172.16.17.0/24 10.10.17.81
# cat /etc/inet/static_routes
# File generated by route(1M) - do not edit.
net 172.16.17.0/24 10.10.17.81
# route -p delete net 172.16.17.0/24 10.10.17.81
delete net 172.16.17.0/24: gateway 10.10.17.81
delete persistent net 172.16.17.0/24: gateway 10.10.17.81
# route -p show
No persistent routes are defined
# cat /etc/inet/static_routes
# File generated by route(1M) - do not edit.
```

Do not edit `/etc/inet/static_routes` directly.

To see how the system would route a packet to a specific address, use `route get`:

```
# route get www.google.com
route to: cf-in-f103.google.com
destination: default
mask: default
gateway: 10.10.17.81
interface: e1000g0
flags: <UP,GATEWAY,DONE,STATIC>
recvpipe sendpipe ssthresh rtt,ms rttvar,ms hopcount mtu expire
0 0 0 0 0 0 1500 0
```

dladm

Linux uses `ethtool` for querying settings of an ethernet device and changing them. OpenSolaris uses the general purpose data link administration tool, `dladm`. Historically, `ndd` was used for tuning network interfaces. Administrators should use `dladm` rather than `ndd` for datalink operations, although `ndd` is still present to support some older interfaces.

For a more in-depth discussion of `dladm` see this paper: <http://opensolaris.org/os/project/brussels/Documentation/wp/>

See Appendix A for a table of `ethtool` capabilities and what Solaris command they map to.

nsswitch

The `nsswitch` mechanism is similar on Linux and OpenSolaris, but many Linux users can ignore it, while OpenSolaris users who manually configure their networks may have to pay attention to how netmasks and hostnames are looked up. Linux network configurations usually specify IP addresses and netmasks; while Solaris configurations can specify IP addresses and netmasks, it is common practice to use names for hosts and gateways (`/etc/defaultrouter`) and to let the system look up netmasks based on the IP addresses.

The `nsswitch` mechanism tells the system whether to look up information in local files, DNS, NIS (YP), etc. The configuration file is:

```
/etc/nsswitch.conf
```

OpenSolaris provides several alternative configuration files:

```
% ls /etc/nsswitch.*
/etc/nsswitch.conf
/etc/nsswitch.files
/etc/nsswitch.nis
/etc/nsswitch.dns
/etc/nsswitch.ldap
/etc/nsswitch.nisplus
```

For the purposes of network configuration, hosts and netmasks are of interest:

```
% grep -E '^hosts|^netmasks' /etc/nsswitch.conf
hosts: files
netmasks: files
```

When OpenSolaris automatically configures the network, DNS is usually set up by DHCP. To set up DNS manually, edit `/etc/resolv.conf` (see `resolv.conf` below) and edit `/etc/nsswitch.conf` to use DNS or copy `nsswitch.dns` to `nsswitch.conf`:

```
% grep ^hosts /etc/nsswitch.conf
hosts: files dns
```

Note: OpenSolaris interprets the above “hosts: files dns” directive to mean “try to resolve the host name through `/etc/hosts` and if that doesn't work, through the dns server”.

By default, netmasks follow the Class A, Class B, and Class C conventions. To use a non-standard netmask, for example 10.10.18.0/24 or 172.16.17.16/28, add these non-standard netmasks to your netmask database. For example, if you are using files:

```
% grep ^netmasks /etc/nsswitch.conf
netmasks: files
```

add these entries to `/etc/netmasks`:

```
10.10.18.0      255.255.255.0
10.10.17.16    255.255.244.240
```

See the `netmask(4)` man page for details.

Resolv.conf

When OpenSolaris automatically configures the network, DNS is usually set up by DHCP. To set up DNS manually, make sure `nsswitch` points to DNS (see `nsswitch` above) and edit `/etc/resolv.conf` to set up DNS. See the `resolv.conf(4)` man page for details. Here is a typical configuration:

```
% cat /etc/resolv.conf
domain mydomain.com
search mydomain.com
nameserver 10.10.17.16
```

Tcpwrappers

On Linux `tcpwrappers`, `tcpd`, is configured for `inetd` services in `/etc/inet.conf` or `/etc/xinetd.conf`. On OpenSolaris it is configured via `inetadm` or `svcadm`. For example, to enable `tcpwrappers` for a single service:

```
# inetadm -m telnet tcp_wrappers=TRUE
#
```

To enable `tcpwrappers` for all `inetd` services:

```
# inetadm -M tcp_wrappers=true
# svcadm refresh inetd
#
```

Or:

```
# svccfg -s inetd setprop defaults/tcp_wrappers=true
# svcadm refresh inetd
#
```

On both Linux and OpenSolaris some services such as `sshd` always check `/etc/hosts.allow` and `/etc/hosts.deny`.

Ipfilters and Iptables

Current Linux distributions use ipfilters to filter and rewrite network packets. OpenSolaris uses ipfilter. See these man pages for details:

- `ipfilter(5)`
- `ipf(4)`
- `ipf(1m)`
- `ipnat(4)`
- `ipnat(1m)`

Network Debugging Tools

Solaris has many of the same debugging tools as does Linux:

```
ping
netstat
traceroute
```

Rather than `tcpdump`, OpenSolaris comes with `snoop` installed; check the man pages for details.

General OpenSolaris System Administration Notes

Root: User (Login) versus Role

By default OpenSolaris does not configure a root account, although a root password is set up during installation. Rather, some users are allowed to assume the "role" of root. The user that was set up during installation is allowed to assume the role of root via the `pfexec` command. To run a command with root privileges:

```
% pfexec <command and arguments>
%
```

To get a root shell:

```
% pfexec bash
#
```

Alternatively, the `su` command with the root password also works:

```
% su -
Password:
#
```

Look at the `pam_roles(5)` and `pam.conf(4)` man pages for instructions on using the `allow_remote` option to allow root access via ssh.

Look at the `-K type=normal` option in the `rolemod(1m)` man page for instructions on creating a normal root account.

Man Pages

Use the `'s'` option to specify a section of the man pages:

```
man -s 1 pkg
man -s 5 pkg
```

OpenSolaris has finer divisions within sections. For example, section 1m is different from section 1:

```
% man -s 1 ifconfig
No entry for ifconfig in section(s) 1 of the manual.
% man -s 1m ifconfig
...
```

The windex database does not come pre-built, which means `man -k` will not work right out of the box, but it is easy to build the windex database:

```
% man -k snoop
/usr/share/man/windex: No such file or directory
% pfexec bash
# catman -w
# exit
exit
% man -k snoop
snoop                snoop (1m)          - capture and inspect network packets
```

IPS: Image Packaging System

On OpenSolaris the `pkg` command combines the roles of `yum` and `rpm`. (On older versions of Solaris the `pkgadd/pkginfo/pkgchk` commands performed the role of `rpm`.)

For example, to install `tcpdump` which is not part of the OpenSolaris base installation, first search for the package:

```
# pkg search -r tcpdump
```

The package is not found at the default publisher (repository), `pkg.opensolaris.org/release`, so search the community contributed publisher, `pkg.opensolaris.org/contrib`:

```
# pkg search -r -s http://pkg.opensolaris.org/contrib tcpdump
```

INDEX	ACTION	VALUE	PACKAGE
info.upstream	set	Tcpdump	pkg:/tcpdump@3.9.8-0.101
basename	file	usr/sbin/tcpdump	pkg:/tcpdump@3.9.8-0.101
pkg.name	set	tcpdump	pkg:/tcpdump@3.9.8-0.101

The package is there, so add `pkg.opensolaris.org/contrib` to the list of publishers:

```
# pkg set-publisher -O http://pkg.opensolaris.org/contrib/
contrib.opensolaris.org
# pkg publisher -H contrib.opensolaris.org http://pkg.opensolaris.org/contrib/ opensolaris.org (preferred) http://pkg.opensolaris.org/release/
```

Install `tcpdump`:

```
# pkg install tcpdump
# pkg contents tcpdump | grep 'tcpdump$'
usr/sbin/tcpdump
```

SMF: Service Management Framework

The Service Management Framework is a superset of the functionality provided by `service(8)` and `chkconfig(8)` on Linux. For general service management see the following:

```
smf(5)ndd
svcs(1)
svcprop(1)
svcadm(1m)
svccfg(1m)
```

For Internet services, see:

```
inetd(1m)
inetadm(1m)
```

See [Enabling Manual Network Configuration](#) above for one use of `svcadm(1m)` in a network context. Here are some more examples.

If you can't remember the exact name of a service, you can search the entire list, in this case for any service that mentions 'nfs'.

```
% svcs -a | grep nfs
disabled      21:12:20  svc:/network/nfs/nlockmgr:default
disabled      21:12:20  svc:/network/nfs/status:default
disabled      21:12:20  svc:/network/nfs/cbd:default
disabled      21:12:20  svc:/network/nfs/mapid:default
disabled      21:12:20  svc:/network/nfs/client:default
disabled      21:12:22  svc:/network/nfs/server:default
disabled      21:12:55  svc:/network/nfs/rquota:default
```

And then enable the desired service.

```
% svcadm enable network/nfs/client:default
```

When you detect a service problem, a good first step for debugging is to quickly scan for services that are enabled but not running or are preventing another enabled service from running.

```
% svcs -xv
```

Most services are disabled by default. To see all network services and whether they are online, disabled, or in maintenance mode (broken), run these two commands:

```
% svcs 'network*'
STATE      STIME      FMRI
disabled   16:53:18   svc:/network/ipfilter:default
...
online     16:53:39   svc:/network/dns/client:default
...
maintenance 16:54:18   svc:/network/rpc/smsserver:default
...
% inetadm
ENABLED    STATE      FMRI
disabled   disabled   svc:/network/telnet:default
...
enabled    maintenance  svc:/network/rpc/smsserver:default
...
enabled    online     svc:/network/security/ktkt_warn:default
...
%
```

Miscellaneous System Configuration and Status

On Linux, system configuration and status can be managed and monitored via:

```
/etc/modprobe.conf
/etc/sysctl.conf
/proc/sys/...
/proc/sys/kernel/...
etc.
```

Here are some of the OpenSolaris interfaces and their man sections:

<code>prstat(1m)</code>	equivalent to <code>top(1)</code>
<code>fmstat(1m)</code>	fault management statistics
<code>cfgadm(1m)</code>	hardware configuration administration
<code>kstat(1m)</code>	kernel statistics
<code>driver.conf(4)</code>	similar to <code>modprobe.conf(5)</code>
<code>/etc/system(4)</code>	similar to <code>/etc/sysctl.conf(5)</code>

Appendix A: Ethtool Equivalent Commands

On Linux, the 'ethtool' is typically used to configure driver parameters. A mapping of equivalent Solaris tools is provided below. The table can be found online at: <http://opensolaris.org/os/project/brussels/Documentation/ethtool-dladm-comparison/>

Property	Linux	OpenSolaris
query pause param	ethtool -h -a	dladm show-linkprop -p flowctrl eth0
change autoneg	ethtool -A eth0 autoneg [on,off]	dladm {set, show, reset} - linkprop -p adv_autoneg_cap eth0
change rx/tx pause params	ethtool -A eth0 rx on/off [tx on/off]	dladm set-linkprop -p flowctrl={tx, rx, no, bi}
interrupt coalescing	ethtool [-c eth0] [-C eth0 xxx]	N/A (automatically tuned in the stack)
rx/tx ring parameter info	ethtool [-g eth0] [-G eth0]	N/A (automatically tuned in the stack)
query driver info	ethtool -i eth0	dladm show-linkprop -p eth0
register dump	ethtool -d eth0	pcitool
eprom dump	ethtool [-e eth0] [-E eth0]	N/A
restart autoneg	ethtool -r eth0	automatically restarted when any of the autonegotiation or advertised speed/duplex props are modified
driver stats	ethtool -S	kstat
adapter selftest	ethtool -t	N/A, OBP might have such func
modify ethernet speed/duplex autoneg	ethtool -s eth0 [speed 10 ,100 ,1000] [duplex half, full] [autoneg on, off]	dladm set-linkprop -p <MII prop> See dladm (1m) and ieee802.3 man page for supported speed/duplex props
modify phy addr	ethtool eth0 phyad x:x:x:x:x	ifconfig eth0 x:x:x:x:x
Wake On LAN	-s eth0 wol xxx	not currently supported
SecureOn (tm) passwd	ethtool -s eth0 sopass	not currently supported
jumbo frame	ifconfig	dladm set-linkprop -p mtu=<value>eth0

opensolaris.com

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN Web sun.com



©2009 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Solaris and OpenSolaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. SunWIN # 487538 Lit # SWWP11968-0 09/09