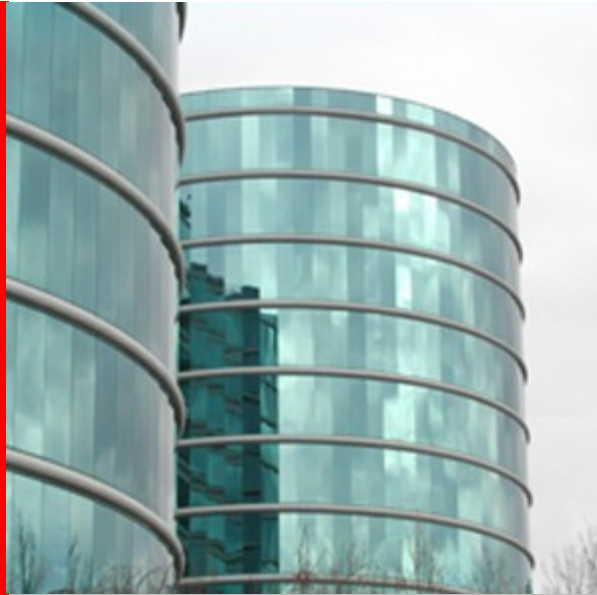


ORACLE®



ORACLE[®]

An Introduction to Solaris 11 Network Configuration

Jon Anderson
Senior Software Engineer – RPE Networking

AGENDA

- Overview – S11 Orientation
- NWAM – Automatic
- NWAM – Manual
- dladm(1M) – data link configuration
- ipadm(1M) – IP interface and address configuration
- Name services – SMF
- Questions?

Network Auto Magic (NWAM)

- Attempt to automate basic network configuration
- Event driven framework to respond to changes in network conditions
- Utilises network configuration profiles (NCP) with activation criteria
- Two system provided profiles by default
 - Automatic (reactive – `Automatic` NCP)
 - Manual (fixed – `DefaultFixed` NCP)
- Possible to create user defined profiles but we're only going to look at the system profiles
- If the `Automatic` or a user defined NCP is active then `netcfg(1M)` and `netadm(1M)` are used

NWAM

- NWAM configuration components consist of:
 - Network Configuration Profiles (NCPs)
 - Network Configuration Units (NCUs) (e.g. link or interface)
 - Location Profiles (name services configuration)
 - External Network Modifiers (ENMs)
 - Known WLANs
- Primary focus is on mobility
- At least one NCP AND one location profile must be active on the system at all times if using automatic network configuration (note, distinct from the `Automatic NCP`)

NWAM

- During a fresh installation the network is configured as follows:
 - For a GUI install, the Automatic NCP is activated
 - For a text installation, options are Automatic, Manual or None
 - Automatic – Automatic NCP is activated
 - Manual – DefaultFixed NCP is activated, configuration options presented
 - None – DefaultFixed NCP is activated but no configuration entered. Only loopback IPv4 and IPv6 will be present on reboot
 - For an AI install, the network is configured according to the AI profile

NWAM

- Automatic mode
 - Created and managed by `nwamd`
 - NCP cannot be modified by the user/admin, location can
 - NCP consists of one link NCU and one interface NCU for each physical link present in the system
 - As links are added/removed from system so are their corresponding NCUs from the `Automatic` NCP
 - Policy is to prefer wired over wireless and plumb IP on all connected wired links (DHCP)
 - One wireless link if no wired links are connected
 - `NoNet` location active until at least one address acquired

NWAM

- Software components:
 - `svc:/network/physical:default`
 - starts/stops profile management daemon (`nwamd`) OR creates/removes persistent `ipadm` configuration
 - SMF property `netcfg/active_ncp` governs this behaviour
 - If `DefaultFixed` then manual `ipadm` configuration
 - Anything else is managed by NWAM and ignores persistent configuration
 - `svc:/network/netcfg:default`
 - starts/stops repository daemon, `netcfgd`

NWAM

- `svc:/network/location:default`
 - Applies the selected location

```
#netadm list
netadm: DefaultFixed NCP is enabled; automatic network
management is not available.
'netadm list' is only supported when automatic network
management is active.

#svcprop -p netcfg/active_ncp svc:/network/physical:default
DefaultFixed

#netadm enable -p ncp Automatic
```

NWAM

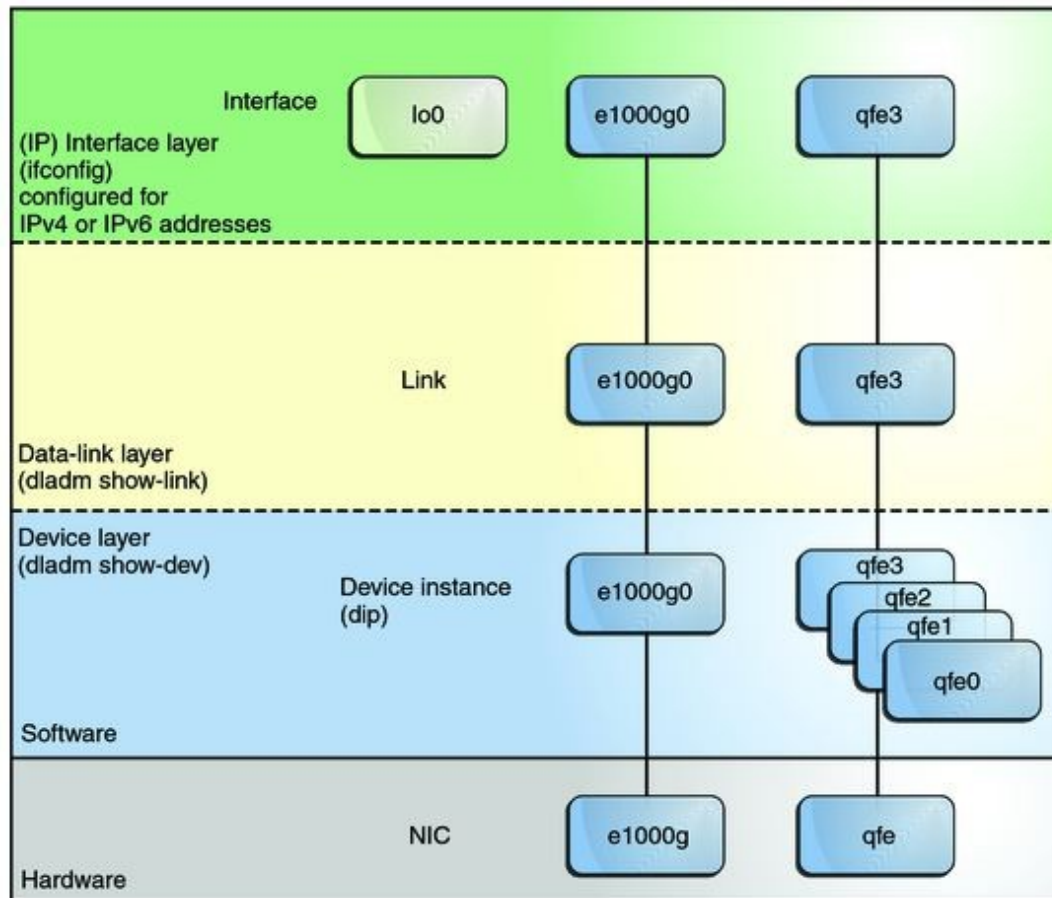
```
#netadm list -x
TYPE          PROFILE      STATE      AUXILIARY STATE
ncp           Automatic   online     active
ncu:phys      net0        online     interface/link is up
ncu:ip        net0        online     interface/link is up
ncu:phys      net1        online     interface/link is up
ncu:ip        net1        offline*   waiting for IP address to be set
ncp           testcfg     disabled   disabled by administrator
loc           Automatic   offline    conditions for activation are unmet
loc           NoNet       offline    conditions for activation are unmet
loc           Lab         online     active
loc           User        disabled   disabled by administrator
```

NWAM

- There is a GUI for NWAM in the form of a GNOME applet
 - GNOME panel applet: `/usr/lib/nwam-manager`
 - GNOME app: `/usr/lib/nwam-manager-properties`
 - Enables display/monitoring and modification of current profiles
 - Displays and facilitates connections to available WLANs
- **IMPORTANT:** If in automatic mode, manual configuration changes to NCP data (links, interfaces) can be undone by `nwamd` if the profile is reactive
- Changes to location data will work but will not persist unless the location profile is changed

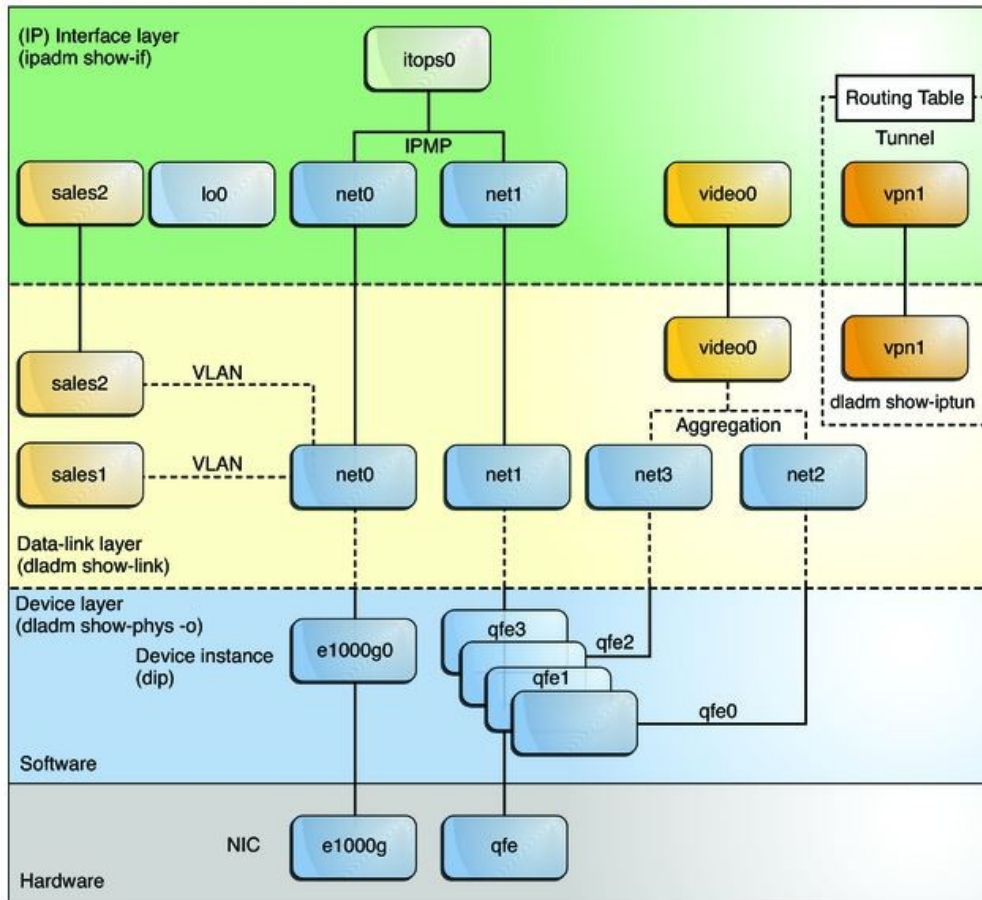
Manual Network Configuration

S10 Network Layout:



Manual Network Configuration

S11 Network Layout:



Manual Network Configuration

dladm(1M)

- Physical interface configuration is managed by `dladm(1M)`
- All L2 instances managed by `dladm`
- Also used for setting/retrieving certain datalink properties
- Harmonizes control over public and private driver properties
- GLDv3 devices (softmac shim for non GLDv3 devices)
- Advanced configuration (VLAN, tunnels, bridging, link aggregation etc.)

Manual Network Configuration

- **Physical** link directly associated with the device and possesses a device name which is, essentially, based on the driver and instance number
- **Datalinks** on top of the physical interfaces can now be given meaningful names
- Abstraction enables simpler dependent network configuration and is easier to deploy
- Persistent configuration across reboots via repository in `/etc/dladm/datalink.conf`
- SMF service
 - `svc:/network/datalink-management:default`

Manual Network Configuration

```
#dladm show-phys  
LINK          MEDIA          STATE          SPEED          DUPLEX          DEVICE  
net0          Ethernet       up             1000           full            nxge0  
net1          Ethernet       down           0              unknown         nxge1  
net2          Ethernet       down           0              unknown         nxge2  
net3          Ethernet       down           0              unknown         nxge3
```

Now also attempts to give a location:

```
#dladm show-phys -L  
LINK          DEVICE          LOC  
net0          nxge0           MB  
net1          nxge1           MB  
net2          nxge2           MB  
net3          nxge3           MB
```


Manual Network Configuration

Displaying link data:

```
#dladm show-link
LINK          CLASS      MTU      STATE    OVER
net0          phys      1500    up       --
net1          phys      1500    down     --
net2          phys      1500    down     --
net3          phys      1500    down     --
vnet0         vnic      1500    up       net0
v1049_net0   vlan      1500    up       net0
```

Manual Network Configuration

Default Generic Link Names

- With a fresh S11 install, generic link names are automatically provided for all the systems physical interfaces
- net# where # is the device instance number
- Physical locality and media type now a consideration when instance numbers are assigned (OBP,SMBIOS)
- Advantages:
 - Easier DR as configuration no longer dependent on name
 - Zone migration more straightforward as link name the same
 - Primary interface generally net0 hence AI config (SC Manifest) can be more generic
 - Data links can now be given contextual names

Manual Network Configuration

Customizing How Generic Link Names Are Assigned

- Only practical **before** installation
- To disable or customize the prefix the SC manifest used by AI needs modifying:

```
<service name="network/datalink-management"
  version="1" type="service">
  <instance name="default enabled="true">
    <property_group name='linkname-policy'
      type='application'>
      <propval name='phys-prefix' type='astring'
        value='net' />
    </property_group>
  </instance>
</service
```

- Can also modify the same property in:

```
svc:/network/datalink-management:default
```

Manual Network Configuration

- Renaming a link:

- Remove any dependent ipadm configuration:

```
# ipadm delete-ip interface
```

- Change the links current name:

```
# dladm rename-link old-linkname new-linkname
```

- In the case of swapping/replacing an interface, you rename the old interface first then re-assign that name to the new interface you want use
- Existing network configuration is preserved
- -t option to `dladm` makes change temporary (e.g. not persistent in datastore)

Manual Network Configuration

- dladm is now used to get/set public and private properties
 - Public properties that can be applied to any driver of the given media type such as link speed, autonegotiation for Ethernet, or the MTU size that can be applied to all datalink drivers
 - Private properties that are particular to a certain subset of NIC drivers for a given media type. These properties can be specific to that subset because they are closely related either to the hardware that is associated with the driver or to the details of the driver implementation itself, such as debugging-related tunables
- GLDv3 dependent

Manual Network Configuration

```
# dladm show-link net0
LINK          CLASS      MTU    STATE    OVER
net0         phys      1500   up       --

# dladm show-linkprop -p mtu net0
LINK    PROPERTY      PERM VALUE      DEFAULT      POSSIBLE
net0    mtu            rw   1500           1500         1500-9194

# dladm set-linkprop -p mtu=9000 net0
# ipadm create-ip net0

# ipadm show-ifprop -p mtu net0
IFNAME      PROPERTY      PROTO PERM  CURRENT      PERSISTENT DEFAULT
POSSIBLE
net0        mtu           ipv4  rw   9000         --          9000
68-9000
net0        mtu           ipv6  rw   9000         --          9000
1280-9000
```

Manual Network Configuration

dladm show-link -s deprecated in favour of dlstat(1M)

```
# dlstat show-link -r net0
  LINK  TYPE      ID  INDEX    IPKTS    RBYTES    INTRS    POLLS    IDROPS
net0   rx      local  --         0         0         0         0         0
net0   rx      other  --         0         0         0         0         0
net0   rx      hw     0  108.62K  19.86M  107.25K   1.37K   4.34K
net0   rx      hw     1   47.77K  48.95M   33.92K  13.86K    406
net0   rx      hw     2   57.44K  73.84M   33.55K  23.89K    375
net0   rx      hw     3   25.00K  14.74M   23.56K   1.44K    254

# dlstat show-link -t net0
  LINK  TYPE      ID  INDEX    OPKTS    OBYTES    ODROPS
net0   tx      local  --         0         0         0
net0   tx      other  --         0         0         0
net0   tx      hw     0  10.95K   1.09M     0
net0   tx      hw     1  12.58K   1.21M     0
net0   tx      hw     2  17.82K   1.37M     0
net0   tx      hw     3   7.09K   823.07K   0
net0   tx      hw     4  24.72K   1.84M     0
net0   tx      hw     5   7.57K   673.17K   0

# dlstat show-link net0
  LINK    IPKTS    RBYTES    OPKTS    OBYTES
net0  251.27K  163.79M   82.58K   7.19M
```

Manual Network Configuration

ipadm(1M)

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] [ipmp] [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

Manual Network Configuration

`ipadm(1M)`

- `ifconfig` suffers from a huge amount of scope creep
- Layering violations
- Most commands are more like an incantation – inconsistent, non-intuitive interface
- Now broken down into stricter layers – `dladm` controls L2 configuration, `ipadm` controls L3 configuration
- `dladm` and `ipadm` also incorporate functionality which was previously delivered by `ndd(1M)`
- `ifconfig` and `ndd` both deprecated

Manual Network Configuration

ipadm(1M)

- ipadm
 - L3 scope
 - Persistent or temporary configuration – including properties
 - Consistent and intuitive command syntax
 - Persistent repository in /etc/ipadm/ipadm.conf
 - Repository managed by /lib/inet/ipmgmtd
 - SMF service
 - `svc:/network/ip-interface-management:default`
 - `ifconfig(1M)` ported to use libipadm interfaces
- Long term goal is to maintain hierarchical object relations with a consistent syntax and datastore
- Not static, transitioning to a more unified model based on SMF

Manual Network Configuration

ipadm(1M)

- EOL of `hostname.<intf>` and `dhcp.<intf>`
 - Awkward administrative interface
 - Error prone
 - No error checking
- `ipmgtmd(1M)` uses the file system event notification API to monitor creation of these files
- `LOG_NOTICE` if `hostname.` or `dhcp.` files detected informing user that `ipadm(1M)` is now the mechanism for delivering persistence

Manual Network Configuration

ipadm(1M) Interface Management

- Divided into four classes: IP, IPMP, Loopback and VNI
 - create-ip [-t] <IP-interface>
 - delete-ip <IP-interface>
 - create-vni [-t] <VNI-interface>
 - delete-vni <VNI-interface>
 - create-ipmp [-t] [-i interface,[...]] <IPMP-interface>
 - delete-ipmp [-f] <IPMP-interface>
- Only libipadm support for creating/deleting a loopback interface

Manual Network Configuration

ipadm(1M) Interface Management

```
# ipadm show-addr
ADDROBJ          TYPE      STATE      ADDR
lo0/v4           static   ok         127.0.0.1/8
lo0/v6           static   ok         ::1/128

# ipadm show-if
IFNAME          CLASS     STATE      ACTIVE OVER
lo0            loopback ok         yes    -

# ipadm create-ip net0
# ipadm create-ip net1
# ipadm show-if
IFNAME          CLASS     STATE      ACTIVE OVER
lo0            loopback ok         yes    --
net0           ip       down      no     --
net1           ip       down      no     --

# ipadm create-addr -T static -a local=192.168.2.1/24 net0/v4
# ipadm show-addr net0/v4primary
ADDROBJ          TYPE      STATE      ADDR
net0/v4primary   static   ok         10.163.210.144/20

# ipadm create-addr -T dhcp net0/dhcp
```

Manual Network Configuration

ipadm(1M) - properties

- Get/set various properties
- Changes can be persistent or temporary using [-t]
- Three general categories
 - `ipadm [show-ifprop | set-ifprop]`
 - Get/set IP interface properties (e.g. MTU)
 - `ipadm [show-addrprop | set-addrprop]`
 - Get/set address properties (e.g. deprecated)
 - `ipadm [show-prop | set-prop]`
 - Get/set generic protocol properties (e.g. tcp SACK)
- Variants include an option to reset property to the default

Manual Network Configuration

ipadm(1M) - properties

```
# ipadm show-ifprop -m ipv4 -p forwarding net0
IFNAME      PROPERTY      PROTO PERM CURRENT  PERSISTENT DEFAULT  POSSIBLE
net0        forwarding    ipv4  rw  off     --       off     on,off

# ipadm set-ifprop -m ipv4 -p forwarding=on net0
# ipadm show-ifprop -m ipv4 -p forwarding net0
IFNAME      PROPERTY      PROTO PERM CURRENT  PERSISTENT DEFAULT  POSSIBLE
net0        forwarding    ipv4  rw  on      on       off     on,off

# ipadm reset-ifprop -m ipv4 -p forwarding net0
# ipadm show-ifprop -m ipv4 -p forwarding net0
IFNAME      PROPERTY      PROTO PERM CURRENT  PERSISTENT DEFAULT  POSSIBLE
net0        forwarding    ipv4  rw  off     --       off     on,off
```

Manual Network Configuration

ipadm(1M) - properties

```
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY          PERM CURRENT      PERSISTENT  DEFAULT      POSSIBLE
tcp  extra_priv_ports    rw  2049,4045      --           2049,4045    1-65535

# ipadm set-prop -p extra_priv_ports+=1500 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY          PERM CURRENT      PERSISTENT  DEFAULT      POSSIBLE
tcp  extra_priv_ports    rw  2049,4045,      1500         2049,4045    1-65535
      1500

# ipadm reset-prop -p extra_priv_ports tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY          PERM CURRENT      PERSISTENT  DEFAULT      POSSIBLE
tcp  extra_priv_ports    rw  2049,4045      --           2049,4045    1-65535
```


ifconfig(1M) → ipadm(1M)

ifconfig(1M)	ipadm(1M)
plumb/unplumb	ipadm create-ip ipadm delete-ip ipadm enable-addr ipadm disable-addr
[<i>address</i> [/ <i>prefix-length</i>] [<i>dest-address</i>]] [addif <i>address</i> [/ <i>prefix-length</i>]] [removeif <i>address</i> [/ <i>prefix-length</i>]][<i>netmask mask</i>] [destination <i>dest-address</i>]{auto-dhcp dhcp} [primary][wait <i>seconds</i>]extend release start	ipadm create-addr -T static ipadm create-addr -T dhcp ipadm show-addr ipadm delete-addr ipadm refresh-addr
[deprecated -deprecated] [preferred -preferred] [private -private] [zone <i>zonename</i> -zones -all-zones][xmit -xmit]	ipadm show-addrprop ipadm set-addrprop ipadm reset-addrprop
up/down	ipadm up-addr ipadm down-addr

Manual Network Configuration

Naming Services

- nsswitch and related files migrated to SMF
 - SMF repository is the primary repository for all naming services configuration
 - The previous behaviour of modifying the name services switch and related files no longer works (up to a point!)
- /etc/nodename – now configured through SMF property config/nodename of `svc:/system/identity:node`
 - If using DHCP, the value provided in the `hostname/nodename` option is preferred

Manual Network Configuration

Naming Services

- Key SMF services (client centric)
 - `svc:/system/name-service/cache:default`
 - `/etc/nscd.conf`
 - `svc:/system/name-service/switch:default`
 - `/etc/nsswitch.conf`
 - `svc:/network/dns/client:default`
 - `/etc/resolv.conf`
 - `svc:/network/nis/client:default`
 - `svc:/network/ldap/client:default`
 - `/var/ldap/*`
- Configuration is achieved via modifying (and refreshing!!!) the appropriate SMF service

Manual Network Configuration

DNS Client Configuration

```
# svccfg
svc:> select dns/client
setprop config/nameserver = net_address: (1.1.1.1 1.1.1.2 1.1.1.3)
setprop config/search = astring: ("uk.oracle.com" "us.oracle.com")
svc:/network/dns/client> select dns/client:default
svc:/network/dns/client:default> refresh
svc:/network/dns/client:default> validate
svc:/network/dns/client:default> select name-service/switch
svc:/system/name-service/switch> setprop config/host = astring: "files dns"
svc:/system/name-service/switch> select name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch:default> validate

# svcadm enable dns/client
# svcadm refresh name-service/switch
# grep host /etc/nsswitch.conf
hosts: files dns

# tail resolv.conf
search uk.oracle.com us.oracle.com
nameserver 1.1.1.1
nameserver 1.1.1.2
nameserver 1.1.1.3
```

Manual Network Configuration

Configuration using nscfg(1M)

- Easier method to import configuration into SMF using nscfg
- e.g. to enable dns/client and modify nsswitch.conf hosts for dns

```
# vi /etc/resolv.conf
# nscfg import -f dns/client
# cp /etc/nsswitch.dns /etc/nsswitch.conf
# nscfg import -f name-service/switch
# svcadm enable dns/client
# svcadm refresh name-service/switch
```

- For LDAP, DefaultFixed and ldapclient(1M)

EOF

- Questions?

Join the UKOUG Solaris SIG



www.ukoug.org/our-communities/solaris



www.facebook.com/solarissig



www.linkedin.com/groups?gid=3010558



[@SolarisSIG](https://twitter.com/SolarisSIG)

ORACLE®