Configuring and Managing Network Components in Oracle® Solaris 11.4

Part No: E60988
November 2020

ORACLE®

Configuring and Managing Network Components in Oracle Solaris 11.4

Part No: E60988

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Référence: E60988

Restrictions de licence/Avis d'exclusion de responsabilité en cas de dommage indirect et/ou consécutif

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Exonération de garantie

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Avis sur la limitation des droits

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Avis sur les applications dangereuses

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Marques

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Inside sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Epyc, et le logo AMD sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Avis d'exclusion de responsabilité concernant les services, produits et contenu tiers

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Date de publication et mention de la version préliminaire de Disponibilité Générale ("Pre-GA")

Version préliminaire de Disponibilité Générale ("Pre-GA") : 15.01.2020

Avis sur la version préliminaire de Disponibilité Générale ("Pre-GA") de la documentation

Si ce document est fourni dans la Version préliminaire de Disponibilité Générale ("Pre-GA") à caractère public ou privé :

Cette documentation est fournie dans la Version préliminaire de Disponibilité Générale ("Pre-GA") et uniquement à des fins de démonstration et d'usage à titre préliminaire de la version finale. Celle-ci n'est pas toujours spécifique du matériel informatique sur lequel vous utilisez ce logiciel. Oracle Corporation et ses affiliés déclinent expressément toute responsabilité ou garantie expresse quant au contenu de cette documentation. Oracle Corporation et ses affiliés ne sauraient en aucun cas être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'utilisation de cette documentation.

Mention sur les informations confidentielles Oracle

INFORMATIONS CONFIDENTIELLES ORACLE. Destinées uniquement à un usage autorisé. Ne pas distribuer à des tiers.

Avis sur la reconnaissance du revenu

Si ce document est fourni dans la Version préliminaire de Disponibilité Générale ("Pre-GA") à caractère privé :

Les informations contenues dans ce document sont fournies à titre informatif uniquement et doivent être prises en compte en votre qualité de membre du customer advisory board ou conformément à votre contrat d'essai de Version préliminaire de Disponibilité Générale ("Pre-GA") uniquement. Ce document ne constitue en aucun cas un engagement à fournir des composants, du code ou des fonctionnalités et ne doit pas être retenu comme base d'une quelconque décision d'achat. Le développement, la commercialisation et la mise à disposition des fonctions ou fonctionnalités décrites restent à la seule discrétion d'Oracle.

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse : http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info ou le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs si vous êtes malentendant.

Contents

Using This Documentation

- **Overview** – Provides information about how to configure and administer various network components in Oracle Solaris, such as datalinks, IP interfaces and addresses, naming and directory services, network profiles, and wireless networks.
- **Audience** – System administrators who are responsible for managing network configuration.
- **Required knowledge** – Basic and advanced network administration concepts and practices.

Product Documentation Library

Documentation and resources for this product and related products are available at https://docs.oracle.com/cd/E37838_01/.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

About Network Administration in Oracle Solaris

This chapter provides an overview of Oracle Solaris network configuration which you perform after installing the operating system. It contains the following topics:

- "What's New in Network Configuration"
- "Description of the Oracle Solaris Network Protocol Stack"
- "Oracle Solaris Network Administration Commands"
- "Required Information to Configure an Oracle Solaris System on the Network"
- "Resources for Network Administration in Oracle Solaris"

For information about configuring the network while installing Oracle Solaris, see *Automatically Installing Oracle Solaris 11.4 Systems* and *Manually Installing an Oracle Solaris 11.4 System*.

What's New in Network Configuration

The following features are new or have changed in this release.

Removal of WEP Support

Support for the Wired Equivalent Privacy (WEP) wireless network security standard has been removed. Consequently, Oracle Solaris WiFi drivers can no longer perform encryption or decryption by using WEP. For more information, see Chapter 6, "Administering Wireless Networks in Oracle Solaris".

SMF Management of Persistent Network Configuration

Persistent network configuration is now managed through the Service Management Facility (SMF). This change aligns network configuration with other system components that use SMF as a core storage repository. It also enables you to apply customized network configuration during an installation by specifying network component settings in system configuration profiles.

In an SMF service, a class of objects having persistent configuration is represented as a property group and shares the name of the class. The classes are as follows:

- Datalinks
- Flows
- IP interfaces and protocol-specific configuration (IP, SCTP, TCP, and ICMP)
- Static IP routes
- Known wireless local area networks (WLANs)
- External network modifiers (ENMs)

Each class or property group would contain children property groups corresponding to the objects in the class. For example, under `datalink` group, you can specify configurations for specific datalink groups on the system, such as a `net0`, `ether0`, `net1`, and so on.

During an installation, you can use system configuration profiles that would implement defined network configuration on target installation clients as shown in the following example:

```
<service name="network/datalink-management" version="1" type="service">
     <instance name="default">
        <property_group name="datalinks" type="application">
            <!-- specify jumbo frames (mtu=9000) for net0 physical datalink -->
            <property_group name="net0" type="datalink-phys">
                <propval name="mtu" type="count" value="9000"/>
            </property_group>
            <!-- specify custom physical datalink name ether0 for e1000g1 device -->
            <property_group name="ether0" type="datalink-phys">
                <propval name="devname" type="astring" value="e1000g1"/>
            </property_group>
        </property_group>
    </instance>
</service>
```

For a summary of changes as a consequence of SMF-managed network configuration, see Appendix A, “Datalink Properties”.

Removal of Network Profiles and Locations

A substantial portion of the functionality of the Network Auto-Magic (NWAM) feature that was introduced in Oracle Solaris 11 has been removed. In particular, network profiles, including the network configuration profile (NCP) and the Location profile, are now unsupported.

- **NCPs and Location profiles** – The removal includes all system-defined NCPs (`Automatic` and `DefaultFixed`) as well as user-define NCPs. In Oracle Solaris 11.4, you configure the network only through available command lines.

  With the removal of Location profiles, the `svc:/network/location` SMF service is removed as well. Note that because the various naming services are not automatically changed, there is no need to save or restore any location-related system-wide network configuration if you previously configured this information by using a Location profile.
- **Known wireless local area networks (WLANs)** – This type of network profile is still supported. However, you no longer administer WLANs by using the `netcfg` command. Instead, you use new `dladm` subcommands to create and manage this type of configuration. This change enables you to continue to use Known WLANs on notebook PCs. See "Administering Known WLANs" on page 92.

  The subcommands that you use to scan and select wireless networks have also been removed. Instead, you use the existing `dladm` subcommands, `scan-wifi` and `connect-wifi`, to connect to wireless networks. The `show-events` subcommand is still supported because it provides a means to listen for events that the `nwamd` daemon sends out.

  For more information, see Chapter 6, "Administering Wireless Networks in Oracle Solaris".
- **External network modifiers (ENMs)** – This type of profile is still supported, as they provide an easy way to run scripts, especially if those scripts have to run automatically whenever certain networking conditions are met. These conditions could include an interface obtaining an IP address or a link state changing.

  You continue to use the `netcfg` command to create and modify ENMs, and the `netadm` command to enable and disable ENMs, as well as check ENM status.

  Because ENMs are the only type of configuration object (or profile) that you configure with the `netcfg` command in Oracle Solaris 11.4, the `enm` keyword and `-p` *profile-type* option has been removed. As a result, the syntax that you use to create an ENM has been simplified, as shown in the following example:

  ```
  netcfg> create myenm

  $ netadm enable myenm
  ```

  For more information, see Chapter 5, "Administering External Network Modifiers in Oracle Solaris".

- **Network configuration during an installation** – The removal of support for NCPs and Location profiles has also changed how the network can be configured during an installation, depending on the installation method that you use.

  Changes for each installation method are as follows:

  - **Text installations** – During a text installation, the `Automatic` and `Manual` options are no longer available. Instead, you must select a network interface to configure and then specify whether to configure the IP address for that interface statically or by using DHCP. The `None` option that enables you to skip networking during an installation and configure the network afterwards is still available.
  - **Automated Installations (AI)** – If you previously used system configuration profiles to specify the `Automatic` NCP for the `netcfg/active_ncp` property of the `svc:/network/physical:default` SMF service, you will need to run the `sysconfig create-profile` command to select which interfaces to configure and then create a new system configuration profile based on the new information.
  - **Upgrading from Oracle Solaris 11 to Oracle Solaris 11.4** – The existing network configuration is upgraded as follows:
    - The existing settings for the currently active NCP and Location profile are retained. However, due to the removal of support for the NWAM reactive mode, the system will behave as though any previously active interfaces were configured to use the `DefaultFixed` NCP upon reboot.
    - Known WLANs are migrated under `dladm` control and will continue to work. However, you must now use `dladm` subcommands rather the `netcfg` command to administer this type of configuration. See Chapter 6, “Administering Wireless Networks in Oracle Solaris”.
- **Network administration GUI (formerly NWAM) is removed** – Support for network administration and the monitoring of network connectivity from the desktop is removed in Oracle Solaris 11.4.

Support for Dynamic MAC Addresses and VLAN IDs

This feature is useful in configuring kernel zones. If you do not know in advance the exact MAC address and VLAN ID for a kernel zone, you can specify the prefixes of allowable MAC addresses and VLAN ID ranges instead. This capability enables a kernel zone to communicate to the global host which MAC address and VLAN ID to use when the system boots.

On kernel zones, the output of the `dladm show-phys -o` command would include the `ALLOWED-ADDRESSES` and `ALLOWED-IDS` columns:

```
$ dladm show-phys -o link,media,device,allowed-addresses,allowed-vids
```

```
LINK    MEDIA       DEVICE   ALLOWED-ADDRESSES   ALLOWED-VIDS
net0    Ethernet    zvnet0   fa:16:3f,           100-199,
                             fa:80:20:21:22      400-498,500
```

The `ADDRESS` column, which displays the MAC address of a physical datalink, is still supported.

See the `dladm`(8) and `zonecfg`(8) man pages.

Naming of Persistent (Static) Routes

A new `-name` option has been added to the `route` command. This option enables you to specify a name when adding, modifying, deleting, or displaying information about a persistent (static) route. In previous releases you could only refer to a route by its destination and gateway. Note that you can only use this option for persistent (static) routes.

For more information, see "Creating Persistent (Static) Routes" in *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer* and the `route`(8) man page.

Support for Aggregation of SR-IOV-Enabled Interfaces

You can now create a SR-IOV-enabled DLMP aggregation by grouping several SR-IOV-enabled NICs and configure SR-IOV VNICs on the SR-IOV-enabled DLMP aggregated ports. Similarly, you can also create a DLMP aggregation by grouping multiple InfiniBand host channel adapter (HCA) ports and configure IPoIB VNICs over this DLMP aggregation. For more information, see DLMP Aggregation of SR-IOV NICs and "DLMP Aggregation of InfiniBand Host Channel Adapter (HCA) Ports" in *Managing Network Datalinks in Oracle Solaris 11.4*.

Support for Port-Based Authentication on Wired Datalinks

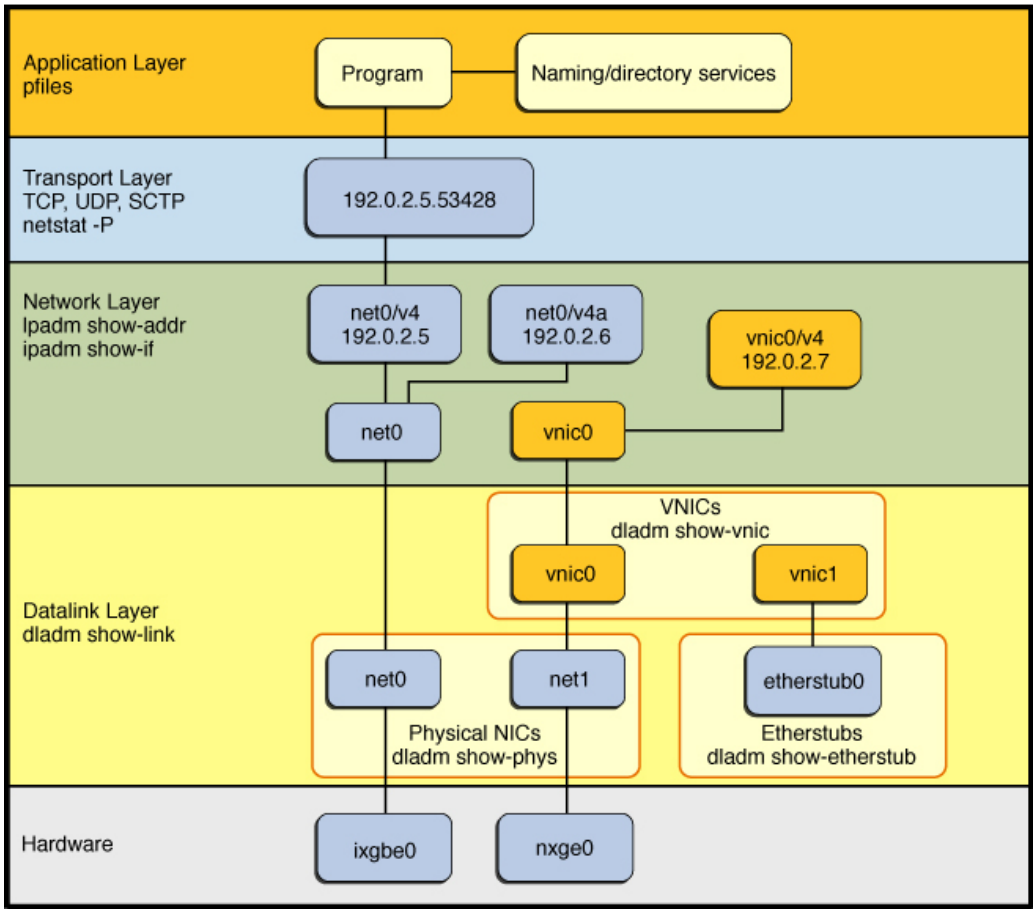
Client-side support for network access control for wired networks through the IEEE 802.1X feature is included in this release. You can use this feature to restrict the use of IEEE 802 LAN service access points (ports) and to secure communications between authenticated devices. See Chapter 3, "Protecting Networks With IEEE 802.1X Certificates" in *Securing the Network in Oracle Solaris 11.4*.

Description of the Oracle Solaris Network Protocol Stack

Network interfaces provide a connection between a system and the network. These interfaces are configured over datalinks, which in turn correspond to instances of hardware devices on the system.

Starting with Oracle Solaris 11, the network configuration on the network layer is no longer bound to the chipset or to the network topology in the hardware layer. Thus, on the datalink layer, the devices are assigned the generic name `net*` rather than device based names.

FIGURE 1 Oracle Solaris 11.4 Network Protocol Stack



This implementation makes network administration more flexible in the following ways:

- Network configuration is insulated from any changes that might occur in the hardware layer. Link and interface configurations are preserved even if the underlying hardware is removed. These same configurations can then be reapplied to any replacement NIC, provided that the two NICs are of the same type.
- With the abstraction of the datalink layer, multiple networking abstractions, or configurations, such as virtual local area networks (VLANs), virtual network interface cards (VNICs), physical devices, link aggregations, and IP tunnels are unified into a common, administrative entity, which is the datalink.

Using Rights Profiles to Perform Network Configuration

Oracle Solaris implements role-based access control (RBAC) to control system access. To perform tasks associated with network configuration, you must be assigned at least the Network Management profile. This profile is a superset that consists of other network-related profiles such as in the following partial list:

- Name Service Management for configuring name services.
- Network Wifi Management for configuring WiFi.
- Elastic Virtual Switch Administration for configuring the elastic virtual switch.
- Network Observability for accessing observability devices.

To obtain a complete list of the profiles in the Network Management profile, type:

```
$ profiles -p "Network Management" info
```

An administrator that has the `solaris.delegate.*` authorization can assign the Network Management profile to users to enable them to administer the network.

For example, an administrator assigns the Network Management rights profile to user `jdoe`. Before `jdoe` executes a privileged network configuration command, `jdoe` must be in a profile shell. The shell can be created by issuing the `pfbash` command. Or, `jdoe` can combine `pfexec` with every privileged command that is issued, such as `pfexec dladm`.

As an alternative, instead of assigning the Network Management profile directly to individual users, a system administrator can create a role that would contain a combination of required profiles to perform a range of tasks.

Suppose that a role `netadmin` is created with the profiles for network configuration as well as zone creation and configuration. User `jdoe` can issue the `su` command to assume that role. All roles automatically get `pfbash` as the default shell.

For more information about rights profiles, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

Oracle Solaris Network Administration Commands

You use three commands to administer network configuration in this Oracle Solaris release. These commands are described in the sections that follow.

`dladm` Command

The `dladm` is used to configure the datalinks on a system as well as manage the following types of network configuration:

- **Physical interfaces** – Ethernet, wireless, and InfiniBand
- **Virtual networking features** – Etherstubs, VNICs, and IP tunnels
- **Switch features** – Link aggregations, VLANs, and bridging technologies
- **Device characteristics** – Speed, duplexing, priority, and feature negotiation

The `dladm` command replaces the `ndd` command that is used to configure protocol properties in previous Oracle Solaris releases.

`ipadm` Command

The `ipadm` command replaces the `ifconfig` command and is used to configured IP interfaces and addresses. Unlike the `ifconfig` command, `ipadm` implements persistent network configuration.

The command also replaces the `ndd` command that was previously used to configure protocol properties.

`route` Command

The `/etc/defaultrouter` file is removed in Oracle Solaris 11.4. Thus, you must use the `route` command to configure persistent and default routes which are then stored in the network routing

tables. See "Creating Persistent (Static) Routes" in *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer*.

`netcfg` and `netadm` Commands

In Oracle Solaris 11.4, the scope of these commands are more limited. The `netcfg` command only manages external network modifiers (ENMs). The `netadm` command is used to enable and disable ENMs and to display information about ENMs and their states.

Required Information to Configure an Oracle Solaris System on the Network

To configure a system's network settings, you need the following basic information:

- Host name
- IP address
- Netmask
- Domain name to which the system belongs
- Default router address

Note - The items are also described in various sections of *Planning for Network Deployment in Oracle Solaris 11.4*.

Depending on the complexity of your network, you might need to specify additional information, such as when configuring the system as a router or as a load balancer. Make sure to refer to the different network administration guides to obtain the requirements.

Resources for Network Administration in Oracle Solaris

See the following references for information about other networking and network administration topics:

- For a quick reference to commonly used network administration commands, see *Oracle Solaris 11.4 Network Administration Cheatsheet*.
- For planning tasks that are required prior to configuring a client system on the network, see *Planning for Network Deployment in Oracle Solaris 11.4*.

- To administer TCP/IP networks and advanced IP features, see Chapter 1, “Administering TCP/IP Networks” in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.
- To learn more about how to secure your network, see *Securing the Network in Oracle Solaris 11.4*.
- To configure a system as a router or a load balancer, see *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer*.
- To configure virtual datalinks, including link aggregations and various types of bridges, see *Managing Network Datalinks in Oracle Solaris 11.4*.
- To implement various network virtualization features, see *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

♦ ♦ ♦ CHAPTER 2

Administering Datalink Configuration in Oracle Solaris

This chapter describes network devices and datalinks and includes tasks for managing basic datalink configuration. It contains the following topics:

- "About Datalink Configuration"
- "Administering Datalink Properties"
- "Customizing Datalink Properties"
- "Additional `dladm` Configuration Tasks"

The tasks that are described in this chapter primarily describe how to configure physical links or links that represent network devices. For other types of network configuration, see:

- *Managing Network Datalinks in Oracle Solaris 11.4*
- *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*

Note - To administer datalink configuration and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 19.

About Datalink Configuration

A datalink represents an object in the second layer (L2) of the Open Systems Interconnection (OSI) model. Datalinks can represent many different L2 entities, such as physical network devices (*physical links*), aggregations of physical datalinks, virtual network interface cards (VNICs), and so on.

When you add network devices to a system, corresponding physical links are automatically created to which are assigned generic names such as `net0`, `net1`, and so on.

The following table illustrates the correspondence between the hardware (NIC), the device instance, the link name, and the interface that is configured over the link. The names of the datalinks are automatically provided by the OS.

Hardware (NIC)	Device Instance	Link's Assigned Name	IP Interface
e1000g	e1000g0	net0	net0
igb	ixgbe	net1	net1

As indicated in this table, while the device instance name remains hardware-based, the datalinks are renamed by the OS after the installation.

The `dladm show-phys` command shows the mapping between datalink names and their corresponding device instances:

```
$ dladm show-phys
LINK     MEDIA        STATE     SPEED     DUPLEX     DEVICE
net2     Ethernet     up        1000      full       bge2
net0     Ethernet     up        1000      full       e1000g0
net3     Ethernet     up        1000      full       nge3
net1     Ethernet     up        1000      full       e1000g1
```

Rules for Valid Link Names

While physical links automatically obtain generic names from the software, you can assign preferred names to other L2 entities that you manually create such as aggregations, VLANs, and others.

The operating system automatically assigns names to physical datalinks. As a best practice, do not revise any of these names. Otherwise, you would introduce complications to future network configuration and maintenance tasks.

For other L2 entities that you create over datalinks, such as aggregations, VLANs, and so on, you can assign preferred names. The names must comply with the following rules:

- Link names must consist of a string and a *physical point of attachment* (PPA) number.
- Link names must abide by the following constraints:
  - Names ideally consist of between 3 to 8 characters. However, names can have a maximum of 31 characters.
  - Valid characters for names are alphanumeric (`a`–`z`, `0`–`9`) and an underscore (_).

Caution - Do not use upper case letters on link names.

- Each datalink must have only one link name at any given time.
- Each datalink must have a unique link name within the system.

For example, a link aggregation might be assigned the name `aggr0`.

Note - You cannot use `lo0` as a link name. It is reserved to identify the IP loopback interface.

Administering Datalink Properties

The `dladm` command is the main command for administering datalinks and their properties.

Displaying General Information About Datalinks

When used without any options, the `dladm` command displays general information about the system's different classes of datalinks. In the following example, aside from physical links, link aggregation (`aggr0`) information is also listed, including its underlying physical links.

```
$ dladm
LINK        CLASS     MTU     STATE     OVER
net0        phys      1500    unknown   --
net1        phys      1500    up        --
net2        phys      1500    unknown   --
net3        phys      1500    unknown   --
net4        phys      1500    up        --
aggr0       aggr      1500    up        net1,net4
```

See *Managing Network Datalinks in Oracle Solaris 11.4* and *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4* for further details.

Displaying a System's Datalinks

You use the `dladm show-link` command to display both the physical and virtual datalinks on a system. A system has as many datalinks as there are installed NICs. You can use various options with this command to customize the information that is displayed.

When used with no additional options or arguments, the `dladm show-link` command displays the following information:

```
$ dladm show-link
LINK                CLASS     MTU    STATE    OVER
net1                phys      1500   down     --
net3                phys      1500   unknown  --
net0                phys      1500   up       --
net2                phys      1500   unknown  --
net11               phys      1500   up       --
net5                phys      1500   up       --
net6                phys      1500   up       --
```

Under the `STATE` column, a datalink's status can be `up`, `down`, or `unknown`.

Use the `-P` option to display persistent configuration information about datalinks. Based on the information that is provided by this command, you can proceed with further network configuration. For example, you can determine the number of NICs on the system, and you can select which datalink to use, over which you can configure IP interfaces. The information that is displayed is similar to the following:

```
$ dladm show-link -P
LINK        CLASS     OVER
net0        phys      --
net1        phys      --
net2        phys      --
```

Displaying the Physical Attributes of Datalinks

Use the `dladm show-phys` command to obtain information about the system's datalinks in relation to the physical NICs with which they are associated. Used without any options, the command displays the following information:

```
$ dladm show-phys
LINK        MEDIA          STATE     SPEED     DUPLEX     DEVICE
net0        Ethernet       up        100Mb     full       e1000g0
net1        Ethernet       down      0Mb       --         nge0
```

```
net2        Ethernet       up        100Mb     full       bge0
net3        InfiniBand     --        0Mb       --         ibd0
```

The previous output displays the following information:

- The corresponding physical NIC of each datalink. For example, `net0` is the datalink name of the NIC `e1000g0`.
- The state of each physical link state helps you identify whether the physical device has connectivity with the external network, if the cable is plugged in and the state of the port on the other end of the cable is `up`.

The `-L` option displays the physical location for each datalink. A datalink's instance number corresponds to its location.

```
$ dladm show-phys -L
LINK     DEVICE     LOCATION
net0     bge0       MB
net2     ibp0       MB/RISER0/PCIE0/PORT1
net3     ibp1       MB/RISER0/PCIE0/PORT2
net4     eoib2      MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

Use the `-m` option to display the MAC addresses of the physical links in a system:

```
$ dladm show-phys -m
LINK                SLOT     ADDRESS            INUSE CLIENT
net0                primary  0:11:22:a9:ee:66   yes   net0
```

This command is similar to using the `ifconfig` command in previous releases.

Display the MAC addresses of the physical and non-physical links in a system as follows:

```
$ dladm show-linkprop -p mac-address
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     mac-address     rw   0:11:22:a9:ee:66 0:11:22:a9:ee:66 0:11:22:a9:ee:66
--
```

Deleting a Datalink

Use the `dladm delete-phys` command to remove a datalink from the system.

Deleting a datalink is only loosely connected to the removal of a physical NIC. For example, if a physical NIC is removed from the system, the datalink configuration that is associated with that NIC remains because the software layer is not bound to the hardware layer. Thus you can still use the datalink configuration on a different underlying physical NIC by assigning the same name to the other NIC's associated link.

If you detach a NIC without replacing it, and you no longer need its datalink configuration, then you can delete the datalink by running the following command:

```
$ dladm delete-phys datalink
```

Tip - To confirm whether a datalink's NIC had been removed, use the `dladm show-phys -P` command. The output provides this information under the FLAGS column, where the `r` flag indicates whether the physical device that is associated with a physical link has been removed.

Obtaining Runtime Statistics for Datalinks

Use the `dlstat` command to obtain runtime datalink statistics for all types of datalinks. When used with no additional options, the `dlstat` displays statistical information about all of the datalinks that are on the system, as shown in the following output:

```
$ dlstat
LINK    IPKTS   RBYTES    OPKTS   OBYTES
net0   58.00K    9.52M    5.61K    1.91M
```

See Chapter 8, “Monitoring Network Traffic and Resource Usage” in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4* and the `dlstat`(8) man page.

Customizing Datalink Properties

You can also use the `dladm` command to set datalink properties and customize them according to the requirements of your network.

The following three `dladm` subcommands are used to administer datalink properties. The properties you can customize depend on what the specific NIC driver supports.

`dladm show-linkprop -p` *propertydatalink*

Displays the properties of a datalink and its current values. If you do not use the `-p` *property* option, then all of the properties of the datalink are displayed. If you do not specify a datalink, then all of the properties for all of the datalinks are displayed.

`dladm set-linkprop -p` *property=value datalink*

Assigns a value to a datalink's property.

`dladm reset-linkprop -p` *property datalink*

Resets a specific property of a datalink to its default value.

Datalink properties that are configurable by using the `dladm` command fall into one of two categories:

- Public properties – Applicable to any driver of the given media type such as link speed, auto-negotiation for Ethernet, or the maximum transmission unit (MTU) size that can be applied to all datalink drivers.
- Private properties – Specific to a subset of NIC drivers for a specific 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.

Enabling Support for Jumbo Frames

The MTU property defines the size of the largest packet that a protocol can transmit from the system. By default, most NIC drivers define the MTU size to 1500. However, for Jumbo frames are traversing the network, the MTU size must be at least 9000.

Note - The MTU property is common to both datalinks and IP interfaces, which means you can have one MTU value for a datalink and another MTU value for the IP interface that is configured over that link. The value of the datalink MTU impacts the possible values that you can set for an IP interface's MTU. For more information, see "Setting the MTU Property" on page 52.

The following example shows how to enable support for Jumbo frames. This example assumes that you have already removed any existing IP interface configuration over the datalink.

```
$ dladm show-linkprop -p mtu net1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net1     mtu             rw   1500         1500         1500      1500

$ dladm set-linkprop -p mtu=9000 net1
$ dladm show-link net1
LINK     CLASS     MTU      STATE     BRIDGE     OVER
web1     phys      9000     up        --         --
```

Modifying Link Speed Parameters

Most network setups include a combination of systems that have varying speed capabilities. These capabilities are usually specified in tandem with duplex settings, for example, full or half-duplex.

Each system advertises speed capabilities to other systems on the network to determine how each system is transmitting and receiving network traffic. Usually, systems use auto-negotiation to determine compatible speed and duplex settings. However, in some instances, it might be necessary to specify a given speed and duplex combination. In these instances, you can set the `speed-duplex` property for the specified datalink. Note that the `speed-duplex` property replaces the legacy en-*speed-duplex*-cap and adv-*speed-duplex*-cap properties (*en1000fdx-cap*) that were used in previous Oracle Solaris releases to set speed-duplex combinations and retrieve their current state, respectively.

You can set the following values for the `speed-duplex` property:

- `10g-f` (10Gigabit full-duplex)
- `1g-f` (1Gigabit full-duplex)
- `1g-h` (1Gigabit half-duplex)
- `100m-f` (100Megabit full-duplex)
- `100m-h` (100Megabit half-duplex)
- `10m-f` (10Megabit full-duplex)
- `10m-h` (10Megabit half-duplex)

The following example shows how to specify 10Gigabit/1Gigabit full-duplex for a datalink:

```
$ dladm set-linkprop -p speed-duplex=10g-f,1g-f datalink
```

To display the new property values, type:

```
$ dladm show-linkprop -p speed-duplex datalink
```

Setting the STREAMS Module on Datalinks

You can set up to eight STREAMS modules to be pushed onto the stream when a datalink is opened. These modules are typically used by third-party networking software such as virtual private networks (VPNs) and firewalls. Documentation about this type of networking software is provided by the software vendor.

The list of modules to push on a specific datalink is controlled by the `autopush` property. You set the value of the `autopush` property by using the `dladm set-linkprop` command.

Modules are pushed on a per-link as well as per-driver basis. If both per-link and per-driver types for the `autopush` configuration exist for a datalink, the per-link information that is set with the `dladm set-linkprop` command is used, and the per-driver information is ignored.

The following example shows how to push the `vpnmod` and `bufmod` modules on top of the link `net0` as follows:

```
$ dladm set-linkprop -p autopush=vpnmod.bufmod net0
```

Obtaining Status Information for Datalink Properties

To obtain status information about datalink properties, you can use either of the following commands:

- `dladm show-linkprop -p` *property datalink*
- `dladm show-ether` *datalink*

Displaying Datalink Properties

To display a complete list of datalink properties, type the command without specifying a property, as shown in the following example:

```
$ dladm show-linkprop net1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net1     speed           r-   10000        10000        10000     --
net1     autopush        rw   --           --           --        --
net1     zone            rw   --           --           --        --
net1     duplex          r-   full         full         full      half,full
     more properties
...
```

Displaying Ethernet Property Values

Use the `dladm show-ether -x` command to obtain a range of information about Ethernet property values:

```
$ dladm show-ether -x net1
```

```
LINK     PTYPE       STATE    AUTO  SPEED-DUPLEX              PAUSE
net1     current     up       yes   1G-f                      both
--       capable     --       yes   1G-fh,100M-fh,10M-fh      both
--       adv         --       yes   100M-fh,10M-fh            both
--       peeradv     --       yes   100M-f,10M-f              both
```

The -x option, displays more than default information and includes the built-in capabilities of the specified link, as well as the capabilities that are currently advertised between the host and the link partner.

In the previous example, the following information is displayed:

- For the Ethernet device's current state, the link is up and functioning at 1 Gigabits per second at full duplex. Its auto-negotiation capability is enabled and has bidirectional flow control, in which both the host and link partner can send and receive pause frames. This information is displayed in the first row of the output.
- Subsequent rows of the output display information about datalink speed capabilities, actual datalink speeds that are advertised, as well as information from the peer system

Additional dladm Configuration Tasks

The following are some network configuration tasks that are simplified by using the dladm command.

How to Move IP Configuration From One Network Device to Another Device

Use the following procedure if you need to preserve the IP configuration that is associated with one network device and then move that configuration to another network device. You might perform this procedure prior to removing a network card from the system or when changing a network cable connection.

The procedure uses e1000g0 and nge0 as sample devices.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See “Using Rights Profiles to Perform Network Configuration” on page 19.

1. **Display the physical link to device mappings on the system.**
   The following example shows that the IP configuration for e1000g0 is down and the configuration needs to be moved to nge0:

```
$ dladm show-phys
LINK    MEDIA    STATE    SPEED    DUPLEX    DEVICE
net0    Ethernet down     0        unknown   e1000g0
net1    Ethernet down     0        unknown   e1000g1
net2    Ethernet up       1000     full      nge0
net3    Ethernet down     0        unknown   nge1
```

2. **Disable the IP configuration on the datalink temporarily, leaving its persistent settings intact.**

   For example:

   ```
   $ ipadm disable-if net0
   ```

3. **Rename the datalink of the device that is down.**

   For example:

   ```
   $ dladm rename-link net0 oldnet0
   ```

   This step makes `oldnet0` as the link name of `e1000g0`.

4. **Assign the primary link name to the datalink that is designated to become the primary device.**

   For example, `net2` is the link name of the `nge0` device. Thus, you would type:

   ```
   $ dladm rename-link net2 net0
   ```

5. **Re-enable the IP configuration on the newly named datalink (`net0`).**

   ```
   $ ipadm enable-if -t net0
   ```

How to Replace a Network Interface Card With Dynamic Reconfiguration

The following procedure applies only to systems that support DR. The procedure specifically refers to configuration steps after DR has completed. You no longer need to reconfigure network links after you complete the DR process. Instead, you just transfer the link configurations of the removed NIC to the replacement NIC.

Note - This procedure does not describe the steps to perform DR itself. Consult your system documentation for that information.

For an introduction to DR, see Chapter 2, "Dynamically Configuring Devices" in *Managing Devices in Oracle Solaris 11.4*.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

Complete the following steps first:

- Ensure that your system supports DR.
- Consult the appropriate manual that describes DR on your system.

  To locate current documentation about DR on Sun servers from Oracle, search for "dynamic reconfiguration" on `https://docs.oracle.com/en/servers/`.

  For information about performing DR in the Oracle Solaris Cluster environment, see *Administering an Oracle Solaris Cluster Configuration* on Oracle Help Center.

1. **Display information about the physical attributes of datalinks and their respective locations on the system.**

   ```
   $ dladm show-phys -L
   ```

2. **Perform the DR process, as described in your system's documentation.**

3. **After you have installed the replacement NIC, do one of the following, depending on the circumstance that applies:**

   - **If you inserted the replacement NIC into the same slot as the old NIC, proceed to Step 4.**

     With the new NIC using the location that was previously occupied by the old NIC, the new NIC inherits the link name and the configuration of the old NIC.

   - **If you inserted the replacement NIC into a different slot, and the new NIC needs to inherit the datalink configuration of the removed NIC, rename the link as follows:**

     ```
     $ dladm rename-link new-datalink old-datalink
     ```

     new-datalink — Refers to the datalink of the replacement NIC that is in a different slot than the location from which the old NIC was removed.

     old-datalink — Refers to the datalink name that is associated with the old NIC that was removed.

     Note - In this scenario, the slot from which the old NIC was removed must remain empty.

For example, the NIC in slot 1 is removed, and then the new NIC is inserted in slot 2. No NIC is inserted in slot 1. Assume that the datalink in slot 1 is `net0`, and the datalink in slot 2 is `net1`. You would specify that the datalink of the new NIC inherit the datalink configuration of the old NIC as follows:

```
$ dladm rename-link net1 net0
```

4. **Complete the DR process by enabling the new NIC's resources so that they are available for use.**

   For example, you can use the `cfgadm` command to configure the NIC. See the `cfgadm`(8) man page.

5. **Display link information.**

   You can use either the `dladm show-phys` command or the `dladm show-link` command to display information about the datalinks.

Example 1 Performing Dynamic Reconfiguration by Installing a New Network Card

The following example shows how a `bge` card with link name `net0` is replaced by an `e1000g` card. The link configurations of `net0` are transferred from `bge` to `e1000g` after `e1000g` is connected to the system.

```
$ dladm show-phys -L
LINK     DEVICE     LOCATION
net0     bge0       MB
net1     ibp0       MB/RISER0/PCIE0/PORT1
net2     ibp1       MB/RISER0/PCIE0/PORT2
net3     eoib2      MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

You can perform the DR-specific steps such as using the `cfgadm` command to remove the `bge` card and then installing the `e1000g` card in its place. After the card is installed, the datalink of `e1000g0` automatically assumes the name `net0` and inherits the configuration of the link.

```
$ dladm show-phys -L
LINK     DEVICE     LOCATION
net0     e1000g0    MB
net1     ibp0       MB/RISER0/PCIE0/PORT1
net2     ibp1       MB/RISER0/PCIE0/PORT2
net3     eoib2      MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2

$ dladm show-link
LINK     CLASS     MTU    STATE    OVER
net0     phys      9600   up       ---
net1     phys      1500   down     ---
```

```
net2     phys      1500   down     --
net3     phys      1500   down     ---
```

SPARC: How to Ensure That the MAC Address of Each Interface Is Unique

Every SPARC based system has a system-wide MAC address, which by default is used by all interfaces. However, some applications require every interface on a host to have a unique MAC address. Certain types of interface configuration such as link aggregations and IP network multipathing (IPMP) similarly require that interfaces must have their own MAC addresses.

The EEPROM parameter `local-mac-address?` determines whether all interfaces on a SPARC based system use the system-wide MAC address or a unique MAC address. The following procedure describes how to use the `eeprom` command to check the current value of the `local-mac-address?` parameter and change it, if necessary.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Determine whether all of the interfaces on the system currently use the system-wide MAC address.**

```
# eeprom local-mac-address?
local-mac-address?=false
```

2. **Change the value of the `local-mac-address?` setting to `true`.**

```
# eeprom local-mac-address?=true
```

When you reboot the system, the interfaces that have factory-installed MAC addresses will use these factory settings rather than the system-wide MAC address. Interfaces without factory-installed MAC addresses will continue to use the system-wide MAC address.

3. **Check the MAC addresses of all of the interfaces on the system.**

Look for cases where multiple interfaces have the same MAC address.

```
$ dladm show-linkprop -p mac-address
LINK PROPERTY    PERM VALUE             EFFECTIVE         DEFAULT           POSSIBLE
net0 mac-address rw   0:14:4f:f9:b1:a9  0:14:4f:f9:b1:a9  0:14:4f:f9:b1:a9  --
net3 mac-address rw   0:14:4f:fb:9a:d4  0:14:4f:fb:9a:d4  0:14:4f:fb:9a:d4  --
net2 mac-address rw   0:14:4f:f9:c:d    0:14:4f:f9:c:d    0:14:4f:f9:c:d    --
net1 mac-address rw   0:14:4f:fa:ea:42  0:14:4f:fa:ea:42  0:14:4f:fa:ea:42  --
```

4. **(Optional) If necessary, manually configure the remaining interfaces so that all interfaces have unique MAC addresses.**

   ```
   $ dladm set-linkprop -p mac-address=mac-address  interface
   ```

   Note - This step is only required if two or more network interfaces have the same MAC address.

5. **Reboot the system.**

♦ ♦ ♦ CHAPTER 3

Configuring and Administering IP Interfaces and Addresses in Oracle Solaris

This chapter describes how to configure IP interfaces for either IPv4 or IPv6 networks. It contains the following topics:

- "About the `ipadm` Command"
- "Configuring IPv4 Interfaces"
- "Configuring IPv6 Interfaces"
- "Customizing IP Interface Properties and Addresses"
- "Customizing IP Address Properties"
- "Disabling, Removing, and Modifying IP Interface Configuration"
- "Monitoring IP Interfaces and Addresses"
- "Migrating From an IPv4 Network to an IPv6 Network"

For information about administering TCP/IP properties such global packet forwarding and transport layer services, see "Administering Transport Layer Services" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Note - To administer IP interfaces configuration and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 19.

About the `ipadm` Command

In Oracle Solaris, the `ipadm` command replaces `ifconfig` as the primary means for configuring IP interfaces. It also replaces the `ndd` command for configuring properties of the following TCP/IP protocols:

- IP

- Address Resolution Protocol (ARP)
- Stream Control Transmission Protocol (SCTP)
- Internet Control Messaging Protocol (ICMP)
- Upper layer protocols such as TCP and the User Datagram Protocol (UDP)

See the `ipadm`(8) man page.

Configuring IPv4 Interfaces

Configuring IP interfaces enables a system to connect to the network. The procedure essentially involves assigning IP addresses to the interfaces.

When you assign an IP address to an IP interface, a corresponding *address object* is created to represent that address.

The address object uses the format *interface*/*address-family*, where *address-family* is either `v4` or `v6`. For example, `net0` with an IPv4 address would have the address object `net0/v4`. Multiple address objects of the same IP interface are distinguished by a letter suffix, such as `net0/v4a`, `net0/v4b`, `net0/v4c`, and so on.

Similarly, an IP interface with IPv6 addresses would have address objects such as `net1/v6a`, `net1/v6b`, and so on.

Any subsequent command on this interface's address can use the address object as reference. For instance, using an address object from the previous examples, you would type `ipadm delete-addr net0/v4b`.

▼ How to Configure an IPv4 Interface

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. Create the IP interface.

```
$ ipadm create-ip interface
```

The interface name follows the datalink name on which the interface is created.

This syntax is the most commonly used to configure networking on a system. However, two other `create` subcommands are available for other types of configuration:

- `create-vni` creates a STREAMS virtual network interface.
- `create-ipmp` creates an IPMP group. See Chapter 2, “About IPMP Administration” in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

2. **Configure the IP interface with a valid IP address.**

   Use one of the following commands depending on the type of address:

   - **Create a static address**

     ```
     $ ipadm create-addr -a IP-address interface
     ```

     The address can be in Classless Inter-Domain Routing (CIDR) notation.

   - **Configure a dynamic address.**

     ```
     $ ipadm create-addr -T dhcp interface
     ```

3. **Display information about the system's IP interfaces.**

   ```
   $ ipadm
   ```

   See also “Monitoring IP Interfaces and Addresses” on page 56 that show how to obtain interface information by using `show-*` subcommands.

4. **If you are configuring a static IP address that uses a host name, add the entries for the IP address to the `/etc/hosts` file.**

   The entries in this file consist of IP addresses and their corresponding host names.

   Note - DHCP configurations do not require updates to the `/etc/hosts` file.

Example 2 Configuring an IPv4 Interface With a Static IP Address

The following example shows how to configure an IPv4 interface with a static IP address. The example also shows how to configure a persistent route for the interface by using the `route` command.

```
$ dladm show-phys
LINK     MEDIA        STATE     SPEED     DUPLEX     DEVICE
net3     Ethernet     up        100Mb     full       bge3

$ dladm show-link
LINK     CLASS     MTU     STATE     OVER
```

```
net3     phys      1500    up        --       --

$ ipadm create-ip net3
$ ipadm create-addr -a 192.0.2.3/24 net3
net3/v4

$ ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
aggr0             ip         down         --         --
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
net3              ip         ok           --         --
   net3/v4        static     ok           --         192.0.2.3/24

$ vi /etc/hosts
# Internet host table
# 127.0.0.1     localhost
203.0.113.14    foohost
192.0.2.3       sales1
```

Example 3 Configuring a Network Interface to Receive an IP Address From a DHCP Server

In the following example, the IP interface is configured to receive its address from a DHCP server. DHCP typically also installs a default route. Therefore, this example does not include a step for manually adding a default route by using the `route` command.

```
$ dladm show-phys
LINK     MEDIA        STATE     SPEED     DUPLEX     DEVICE
net3     Ethernet     up        100Mb     full       bge3

$ dladm show-link
LINK     CLASS     MTU     STATE     OVER
net3     phys      1500    up        --       --

$ ipadm create-ip net3
$ ipadm create-addr -T dhcp net3
net3v4

$ ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
aggr0             ip         down         --         --
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
net3              ip         ok           --         --
   net3/v4        static     ok           --         203.0.113.3/24
```

Configuring IPv6 Interfaces

Configuring an IPv6 interface consists of assigning an IPv6 address to the interface.

Typically, you would use autoconfiguration to configure IPv6 on IP interfaces. Autoconfiguration assigns a link-local address and discovers prefixes and routers that are in use on the subnet. Interfaces with autoconfigured addresses also automatically request DHCPv6 address information.

If the interface is on the same link as a router that currently advertises an IPv6 prefix, the interface obtains that site prefix as part of its autoconfigured addresses.

The identifier portion of an autoconfigured IPv6 address is based on the MAC address of the interface's NIC card. Thus, autoconfiguration works best if the NIC cards are not often replaced. Otherwise, the MAC portion constantly changes as well. The instability would impact performance if other parts of the network infrastructure use stored IP addresses and therefore require stable addresses. In these cases, autoconfiguration might not be optimal.

As an alternative, you can use static addresses instead. Or, you can create IPv6 tokens as explained in “Configuring an IPv6 Token” on page 47. In both cases, the IPv6 addresses remain constant even if the interfaces are changed.

▼ How to Configure a System for IPv6

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See “Using Rights Profiles to Perform Network Configuration” on page 19.

1. **(Optional) If it doesn't yet exist, create the IP interface over the datalink.**

   For example, to create the `net0` IP interface, you would type:

   ```
   $ ipadm create-ip net0
   ```

2. **Assign the IP address or addresses.**

   Use one of the following commands depending on the type of address:

   - **Use autoconfiguration.**

     ```
     $ ipadm create-addr -T autoconfig interface
     ```

   - **Create a static address**

```
$ ipadm create-addr -a IPv6-address  interface
```

Note - Use the same command to add IPv6 addresses after assigning one autoconfigured address to an interface.

3. **(Optional) Create an `/etc/inet/ndpd.conf` file that defines the parameters for the interface variables that are on the node.**

   To create temporary addresses for the host's interface, see "Using Temporary Addresses for an IPv6 Interface" on page 44. For details about the `/etc/inet/ndpd.conf` file, see the `ndpd.conf(5)` man page.

4. **Display the status of the IP interfaces with their IPv6 configurations.**

   ```
   $ ipadm show-addr
   ```

Example 4 Enabling an IPv6 Interface After an Installation

The following example shows how to enable IPv6 on the `net0` interface.

```
$ ipadm show-addr
ADDROBJ   TYPE     STATE   ADDR
lo0/v4    static   ok      127.0.0.1/8
net0/v4   static   ok      203.0.113.74/24

$ ipadm create-addr -T addrconf net0
$ ipadm create-addr -a 2001:db8:3c4d:15::203/32 net0

$ ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8
net0/v4      static     ok      172.16.27.74/24
net0/v6      addrconf   ok      2001:db8:203:baff:fe13:14e1/32
lo0/v6       static     ok      ::1/128
net0/v6a     static     ok      2001:db8:3c4d:15::203/32
```

Using Temporary Addresses for an IPv6 Interface

Temporary addresses cause interfaces to become anonymous, such as the interfaces of a host that needs to access public web servers. Temporary addresses implement IPv6 privacy enhancements as described in *Privacy Extensions for Stateless Address Autoconfiguration in IPv6* (`http://www.rfc-editor.org/rfc/rfc3041.txt`).

Unlike a standard, autoconfigured IPv6 address, a temporary address consists of a 64-bit subnet prefix and a randomly generated 64-bit number. This random number becomes the interface ID segment of the IPv6 address. A link-local address is not generated with the temporary address as the interface ID.

You enable temporary addresses by configuring variables in the `/etc/inet/ndpd.conf` file, as shown in the following procedure.

▼ How to Configure a Temporary IPv6 Address

The `/etc/inet/ndpd.conf` file contains statements that define values for the different time variables for the temporary addresses. Follow these general rules when adding statements in the file:

- To apply a definition to all the interfaces in the system, use this statement syntax:

  `ifdefault` *variable value*
- To apply a definition to a specific interface, use this statement syntax:

  `if` *interface variable value*
- For the variables, the default unit of time is seconds. Simply provide the numeric value. To specify time in hours and days, use *n*`h` (`40h`) and *n*`d` (`30d`), respectively.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Edit the `/etc/inet/ndpd.conf` file by configuring variables related to temporary addresses.**

   a. **Enable generation of temporary addresses.**

      `ifdefault|if` *interface* **`TmpAddrsEnabled true`**

   b. **(Optional) Specify the valid lifetime for the temporary address.**

      `ifdefault|if` *interface* **`TmpValidLifetime`** *duration*

      The default valid lifetime is 7 days (`7d`).

   c. **(Optional) Specify a preferred lifetime for the temporary address, after which the address is deprecated.**

      `ifdefault|if` *interface* **`TmpPreferredLifetime`** *duration*

The default preferred lifetime is one day (`1d`).

Note - Default address selection gives a lower priority to deprecated IPv6 addresses than non-deprecated addresses. A non-deprecated address could be the automatically generated IPv6 address or possibly the interface's IPv4 address. See "Administering Default Address Selection" on page 48.

---

d. (Optional) Specify the lead time in advance of address deprecation, during which the host should generate a new temporary address.

`ifdefault|if` *interface* **`TmpRegenAdvance`** *duration*

The default value is 5 seconds (`5`).

2. Enable the `svc:/network/routing/ndp:default` SMF service.

```
$ svcadm restart ndp:default
```

3. Verify that temporary addresses have been created.

```
$ ipadm show-addr -o all
```

Example 5 Displaying Whether Temporary Addresses Are Enabled

The following example shows the output of the `ipadm show-addr` command after temporary addresses are created. The sample output here is an extract and only IPv6-related information is included.

```
$ ipadm show-addr -o all
ADDROBJ           TYPE     STATE        CURRENT PERSISTENT ADDR

lo0/v4            static   ok           U------ U--        127.0.0.1/8

net0/v4           dhcp     ok           U----D- U--        203.0.113.225/24

lo0/v6            static   ok           U------ U--        ::1/128

net0/v6           addrconf ok           U------ U--
 2001:db8:214:4fff:fef9:b1a9/32
net0/v6           addrconf ok           U--t--S ---
 2001:db8:414:60bb:815c:f4f7:8487:95c2/32
```

Temporary addresses are flagged with `t` under the `CURRENT` column. The `D` flag indicates an IP address that was configured as a result of DHCP negotiation. The `S` flag indicates an address that was configured as a result of IPv6 stateless address autoconfiguration.

See Also

- To set up name service support for IPv6 addresses, see Chapter 4, "Administering Naming and Directory Services on an Oracle Solaris System".
- To configure IPv6 addresses for a server, see "How to Configure a User-Specified IPv6 Token" on page 47.
- To monitor activities on IPv6 nodes, see Chapter 1, "Administering TCP/IP Networks" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Configuring an IPv6 Token

The 64-bit interface ID of an IPv6 address is also referred to as a *token*. By default, autoconfiguration creates the token based on the interface's MAC address.

However, you can create a token that is not based on the MAC address. Instead, you create a 64-bit hexadecimal number as the interface ID. This token remains assigned to the interface, even when a card is replaced.

IPv6 tokens are another alternative to use on systems whose interfaces are routinely swapped. Just like static addresses, they can serve as alternatives to using normal autoconfiguration. See "Configuring IPv6 Interfaces" on page 43.

▼ How to Configure a User-Specified IPv6 Token

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Verify that the interface that you want to configure with a token exists and that no IPv6 addresses are configured on that interface.**

2. **Configure each interface that has a user-specified interface ID (token).**

   `$ ipadm create-addr -T addrconf -i` *token interface*

   The token you create must be in the following format:

   xxxx:*xxxx*:*xxxx*:*xxxx*

   Note - After you create the address object with the token, you can no longer modify the token.

Example 6 Configuring a User-Specified Token on an IPv6 Interface

The following example shows how to configure `net0` with an IPv6 address and a token.

```
$ ipadm show-if
IFNAME   CLASS      STATE    ACTIVE     OVER
lo0      loopback   ok       yes        ---
net0     ip         ok       yes        ---

$ ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8

$ ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
$ ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v6       static     ok      ::1/128
net0/v6      addrconf   ok      2001:db8:1a:2b:3c:4d/10
net0/v6a     addrconf   ok      2001:db8:39f0:1:1a:2b:3c:4d/64
```

After the token is configured, the address object `net0/v6` has both a link-local address, as well as an address with `1a:2b:3c:4d` configured for its interface ID.

See Also

- To update the naming services with the IPv6 addresses of the server, see Chapter 4, “Administering Naming and Directory Services on an Oracle Solaris System”.
- To monitor server performance, see Chapter 1, “Administering TCP/IP Networks” in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Administering Default Address Selection

A single IP interface can have multiple IP addresses. For example, technologies such as IPMP enable multiple NICs to connect to the same IP link layer. That link can have one or more IP addresses. Additionally, IPv6-enabled interfaces have a link-local IPv6 address, at least one IPv6 routing address, and an IPv4 address for at least one interface.

In such cases, applications must know which address to use especially when communicating with target systems. Address selection and ordering is the process by which the operating system selects and prioritizes the appropriate addresses to use to reach destination hosts.

The `/etc/inet/ipaddrsel.conf` file contains selected and prioritized addresses:

TABLE 1 IPv6 Address Selection Policy Table

Prefix	Precedence	Definition
`::1/128`	50	Loopback
`::/0`	40	Default

Prefix	Precedence	Definition
`2002::/16`	30	6to4
`::/96`	20	IPv4 Compatible
`::ffff:0:0/96`	10	IPv4

In the preceding table, IPv6 prefixes (`::1/128` and `::/0`) take precedence over 6to4 addresses (`2002::/16`) and IPv4 addresses (`::/96` and `::ffff:0:0/96`). Therefore, by default, the kernel selects the global IPv6 address of the interface for packets going to another IPv6 destination. The IPv4 address of the interface has a lower priority, particularly for packets going to an IPv6 destination. Given the selected IPv6 source address, the kernel also uses the IPv6 format for the destination address.

Both IPv4-only and dual-stack IPv4/IPv6 systems must perform default address selection. In most circumstances, you do not need to change the default address selection mechanisms in this file. However, you might need to change the priority of address formats in certain cases such as the following:

- If the system has an interface that is used for a 6to4 tunnel, you can assign a higher priority to the 6to4 addresses.
- If you want a particular source address to be used only in communications with a particular destination address, you can add these addresses to the policy table. Then, you can use the `ipadm` command to flag these addresses, as preferred. See the `ipadm`(8) man page.
- If you want IPv4 addresses to take precedence over IPv6 addresses, you can change the priority of `::ffff:0:0/96` to a higher number.
- If you need to assign a higher priority to deprecated addresses, you can add the deprecated address to the policy table. For example, site-local addresses are now deprecated in IPv6. These addresses have the prefix `fec0::/10`. You can change the policy table to assign a higher priority to site-local addresses.

For more information about address selection, see the `ipaddrsel`(8) and `getaddrinfo`(3C) man pages. For IPv6, see the `inet6`(3C) man page.

▼ How to Administer the IPv6 Address Selection Policy Table

Caution - Do not change the IPv6 address selection policy table unless necessary. Otherwise, network problems might occur due to a badly constructed policy table.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Review the current IPv6 address selection policy table.**

```
$ ipaddrsel
# Prefix                  Precedence Label
::1/128                          50 Loopback
::/0                             40 Default
2002::/16                        30 6to4
::/96                            20 IPv4_Compatible
::ffff:0.0.0.0/96                10 IPv4
```

2. **Make a backup copy of the default address policy table.**

   Do one of the following steps:

   - **To make persistent changes:**

     ```
     $ cp /etc/inet/ipaddrsel.conf /etc/inet/ipaddrsel.conf.orig
     ```

   - **To make changes only for the current session:**

     ```
     $ cp /etc/inet/ipaddrsel.conf /etc/inet/temp.conf
     ```

3. **Edit the file by adding customizations in the following format:**

   prefix/prefix-length precedence label [# comment]

   For example, you edit either `ipaddrsel.conf` or `temp.conf`.

4. **Load the modified policy table into the kernel.**

   ```
   $ ipaddrsel -f /etc/inet/filename
   ```

   For example, you load either `ipaddrsel.conf` or `temp.conf`. For non-persistent changes, the kernel uses the policies in `temp.conf` until you reboot the system.

5. **If the modified policy table has problems, restore the default IPv6 address selection policy table.**

   ```
   $ ipaddrsel -d
   ```

Example 7 Modifying the Default IPv6 Address Selection Policy Table

This list illustrates possible ways to change the address priority in the policy table:

- Assign the highest priority to 6to4 addresses.

  ```
  2002::/16                        50 6to4
  ```

```
::1/128                         45 Loopback
```

The 6to4 address format now has the highest priority, 50. Loopback, which previously had a 50 precedence, now has a 45 precedence. The other addressing formats remain the same.

- Designate a specific source address to be used in communications with a specific destination address.

```
::1/128                         50 Loopback
2001:1111:1111::1/128           40 ClientNet
2001:2222:2222::/48             40 ClientNet
::/0                            40 Default
```

This particular entry is useful for hosts with only one physical interface. Here, `2001:1111:1111::1/128` is preferred as the source address on all packets that are bound for destinations within network `2001:2222:2222::/48`. The 40 priority gives higher precedence to the source address `2001:1111:1111::1/128` than to other address formats configured for the interface.

- Favor IPv4 addresses over IPv6 addresses.

```
::ffff:0.0.0.0/96               60 IPv4
::1/128                         50 Loopback
.
```

The IPv4 format `::ffff:0.0.0.0/96` has its precedence changed from the default 10 to 60, the highest priority in the table.

Customizing IP Interface Properties and Addresses

Like datalinks, IP interfaces also have properties that you can customize for your specific network environment. For each interface, two sets of properties exist, one set for IPv4 and the another set for IPv6.

To manage IP interface properties, three `ipadm` subcommands are available:

- `show-ifprop [-p` *property*`] [`*interface*`]` – Displays a property of an IP interface and its current value. Omitting the `-p` *property* option will list all the properties of the specific IP interface. If you do not specify an IP interface, then all the properties of all the IP interfaces are listed.
- `set-ifprop -p` *property=value interface* – Assigns a value to the IP interface's property.
- `reset-ifprop -p` *property interface* – Resets the specific property to its default values.

Setting the MTU Property

Some properties, including the MTU property, are common to both datalinks and IP interfaces. Thus, you can have one MTU value for a datalink and a different MTU value for the interface that is configured over that link. In addition, you can have different MTU values that apply to the IPv4 and IPv6 packets that traverse that IP interface.

When setting MTU properties for an IP interface, keep the following key points in mind:

- The value of the MTU setting of an IP interface cannot be larger than the value of the MTU setting of a datalink. In such cases, the `ipadm` command displays an error message.
- If an IP interface's MTU value is different from the datalink's MTU value, IP packets are limited to the MTU value of the IP interface. For example, if a datalink has an MTU value of 9000 bytes and an IP interface as an MTU value of 1500 bytes, IP packets are limited to 1500 bytes. However, other Layer 3 protocols that are using the underlying Layer 2 protocol can send packets up to 9000 bytes.

  For instructions on customizing datalink properties, including information about how the MTU setting of a datalink impacts the MTU setting of an IP interface, see "Customizing Datalink Properties" on page 28.

Enabling Packet Forwarding

On a network, a host system can receive data packets that are destined for another host. By enabling packet forwarding on the receiving local system, that system can forward the data packet to the destination host. This process is referred to as *IP forwarding* and is disabled by default in Oracle Solaris.

Packet forwarding is managed by a property that you can set on both IP interfaces, as well as for the TCP/IP protocol. If you want to be selective about how packets are forwarded, you can enable packet forwarding on the IP interface. For example, you might have a system that has multiple NICs, where some NICs are connected to the external network, while other NICs are connected to a private network. You can therefore enable packet forwarding only on some of the interfaces, rather than on all of the interfaces.

You can also enable packet forwarding globally on a system by setting the property of the TCP/IP protocol. See "Enabling Global Packet Forwarding" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Note - The `forwarding` property of either IP interfaces or protocols is not mutually exclusive. You can set the property for the interface *and* the protocol at the same time. For example, you could enable packet forwarding globally on the protocol, and then customize packet forwarding for each IP interface on the system. Thus, although enabled globally, packet forwarding can still be selective for the system.

For example, you can enable packet forwarding on an IP interface as follows:

```
$ ipadm set-ifprop -p forwarding=on -m protocol-version  interface
```

where *protocol-version* is either IPv4 or IPv6. You must run this command separately for IPv4 and IPv6 packets.

The following example shows how you might enable only IPv4 packet forwarding on a system:

```
$ ipadm show-ifprop -p forwarding net0
IFNAME   PROPERTY     PROTO   PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
net0     forwarding   ipv4    rw     off       off          off       on,off
net0     forwarding   ipv6    rw     off       --           off       on,off

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

Customizing IP Address Properties

The `ipadm` command enables you to manage IP address-specific properties.

You can customize IP address properties to manage the following network configuration parameters:

- Netmask length
- Whether an IP address can be used as a source address for outbound packets
- Whether the IP address belongs to a global or non–global zone
- Whether the IP address is a private address

Use the following `ipadm` subcommands when working with IP address properties:

- `show-addrprop -p` *property addrobj* – Displays address properties, depending on the options that you use.

  To display the properties of all the IP addresses, do not specify a property or an address object. To display the values of a single property for all the IP addresses, specify only that

property. To display all the properties of a specific address object, specify just the address object.

- `set-addrprop -p` *property=value addrobj* – Assigns values to address properties. Note that you can only set one address property at a time.
- `reset-addrprop -p` *property addrobj* – Restores any default values to the address property.

Note - If you want to change the IP address of a specific interface, do not use the `set-addressprop` subcommand. Instead, delete the address object and create a new one with the new IP address. See "Modifying an IP Interface Configuration" on page 55.

As an example, suppose you want to change the netmask of an IP address. The IP address is configured on the IP interface `net3` and is identified by the address object name `net3/v4`. The following example shows how to revise the netmask:

```
$ ipadm show-addr
ADDROBJ     TYPE       STATE     ADDR
lo0/?       static     ok        127.0.0.1/8
net3/v4     static     ok        198.51.100.3/24

$ ipadm show-addrprop -p prefixlen net3/v4
ADDROBJ  PROPERTY   PERM  CURRENT  PERSISTENT  DEFAULT   POSSIBLE
net3/v4  prefixlen  rw    24       24          24        1-30,32

$ ipadm set-addrprop -p prefixlen=8 net3/v4
$ ipadm show-addrprop -p prefixlen net3/v4
ADDROBJ  PROPERTY   PERM  CURRENT  PERSISTENT  DEFAULT   POSSIBLE
net3/v4  prefixlen  rw    8        24          24        1-30,32
```

Disabling, Removing, and Modifying IP Interface Configuration

This section discusses various ways to manage interfaces you no longer use.

Removing an IP Interface Configuration

Use the `delete-ip` subcommand to remove a configured IP interface. This command is particularly important when you are performing certain datalink configuration tasks. For example, renaming a datalink fails if there are IP interfaces configured over that datalink.

Before attempting to rename the datalink, you would need to use the `ipadm delete-ip` command to remove the existing IP configuration.

For example, you can use the following commands to perform this task:

```
$ ipadm delete-ip interface
$ dladm rename-link old-name new-name
$ ipadm create-ip interface
$ ipadm create-address parameters
```

See "How to Move IP Configuration From One Network Device to Another Device" on page 32 for procedure that includes renaming a datalink. For instructions to reconfigure an IP interface after a datalink has been renamed, see "How to Configure an IPv4 Interface" on page 40.

Disabling an IP Interface Configuration

By default, an IP interface is plumbed and becomes part of the active configuration when you create the interface by using the `ipadm create-ip` command. The interface is flagged as `UP` when the first address is created on the interface.

To remove an interface from active configuration without destroying the configuration, use the `disable-if` subcommand. This subcommand unplumbs the interface in the kernel.

```
$ ipadm disable-if -t interface
```

You can make an IP interface operational and its flag displayed as `UP` as follows:

```
$ ipadm enable-if -t interface
```

Tip - To display the current status of IP interfaces, see "Obtaining Information About IP Interfaces" on page 58.

Modifying an IP Interface Configuration

The `ipadm delete-addr` command deletes a specific address configuration from an IP interface. This command is useful when you simply want to remove an IP address from the system or as part of changing an IP address that is configured on an interface. If you want to change the IP address that is configured on an interface, then you must first remove the original address configuration before assigning a new address configuration. See "How to Modify an Existing IP Address" on page 56.

For instructions on creating an IP address for an interface, see “How to Configure an IPv4 Interface” on page 40.

Note - An interface can have multiple IP addresses. Each address is identified by an address object. To ensure that you are removing the correct address, you must know the address object. Use the `ipadm show-addr` command to display the IP addresses that are configured on an interface. For more information about displaying IP addresses, see “Obtaining Information About IP Addresses” on page 59.

▼ How to Modify an Existing IP Address

The following procedure describes the steps for reconfiguring a system's existing IP address.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See “Using Rights Profiles to Perform Network Configuration” on page 19.

1. **Delete the address object that represents the IP address that you want to reconfigure.**

   ```
   $ ipadm delete-addr addrobj
   ```

2. **Assign a new IP address by using the same address object name.**

   ```
   $ ipadm create-addr -a IP-address addrobj
   ```

   To add another interface to the system, see “How to Configure an IPv4 Interface” in *Configuring and Managing Network Components in Oracle Solaris 11.4*.

3. **(Optional) If necessary, modify the system's host name as follows:**

   ```
   $ hostname new-hostname
   ```

4. **If necessary, modify the entries in the `/etc/hosts` file.**

5. **Reboot the system for the changes to take effect.**

Monitoring IP Interfaces and Addresses

The `ipadm` command has several `show*` subcommands to monitor IP interfaces:

- `show-if` displays interface information.

- `show-addr` provides IP address information.
- `show-ifprop` shows information about interface properties.
- `show-addrprop` lists information about IP address properties.

See the `ipadm`(8) man page.

Obtaining General Information About IP Interfaces

The `ipadm` command provides a comprehensive picture of the system's interfaces. Without accompanying subcommands, `ipadm` displays default information about all of the system's IP interfaces:

```
$ ipadm
NAME        CLASS/TYPE STATE    UNDER  ADDR
lo0         loopback   ok          --  --
  lo0/v4    static     ok          --  127.0.0.1/8
  lo0/v6    static     ok          --  ::1/128
net0        ip         ok          --   --
  net0/v4   dhcp       ok          --  203.0.113.233/23
  net0/v6   addrconf   ok          --  2001:db8:214:4fff:fef9:b1a9/32
  net0/v6   addrconf   ok          --  2001:db8:414:60bb:214:4fff:fef9:b1a9/32
  ipmp0     ipmp       degraded    --   --
  ipmp0/v6  static     ok          --  2001:db8:1:2::4c08/128
  net1      ip         failed   ipmp0  --
  net1/v6   addrconf   ok          --  2001:db8:124:4fff:fe58:1831/32
  net2      ip         ok       ipmp0  --
  net2/v6   addrconf   ok          --  2001:db8:214:4fff:fe58:1832/10
  iptun0    ip         ok          --  --
  iptun0/v4  static    ok          --  198.51.110.5->198.51.110.75
  iptun0/v6  static    ok          --  2001:db8:10:5->2001:db8:223:75
  iptun0/v6a static    ok          --  2001:db8:1a0:7::10:5->2001:db8:7a82:64::223:75
```

The output shows the following information:

- IP interfaces and their associated address objects. The address objects are indented.
- Class of each interface and the types of address of their respective address objects.
- State of each interface and their address objects. An interface can be either a stand-alone or an underlying interface of another type of interface configuration. Configurations that have underlying interfaces are identified in the `UNDER` column. In the example, `ipmp0` is created over `net1` and `net2`. Note, too, that because `net1` has a failed status, then `ipmp0`'s state is also degraded.
- Actual addresses of their respective address objects.

  A pair of addresses in an address object indicates a tunnel configuration.

Obtaining Information About IP Interfaces

To display information about IP interfaces, use the `ipadm show-if` *interface* command. If you do not specify an interface, the information covers all of the interfaces on the system.

```
$ ipadm show-if
IFNAME       CLASS        STATE     ACTIVE     OVER
lo0          loopback     ok        yes        --
net0         ip           ok        yes        --
net1         ip           ok        yes        --
tun0         ip           ok        yes        --
```

The output displays some of the following information:

- Class of each interface, which can be `ip`, `ipmp`, `vni`, or `loopback`.
- State of each interface, which can be `ok`, `offline`, `failed`, `down`, or `disabled`.

  `failed` applies to an IPMP group that can no longer host traffic because the underlying interfaces have failed. `disabled` refers to the IP interface that is unplumbed by using the `ipadm disable-if` command.
- The `OVER` column applies to IPMP groups. It identifies the IPMP group that uses the listed IP interfaces.

Obtaining Information About IP Interface Properties

Use the `ipadm show-ifprop` *interface* command to obtain information about the properties of IP interfaces. If you do not specify a property or an interface, then information about all the properties of all the IP interfaces on the system is displayed.

```
$ ipadm show-ifprop -p mtu net1
IFNAME  PROPERTY  PROTO  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
net1    mtu       ipv4   rw    1500      --         1500     68-1500
net1    mtu       ipv6   rw    1500      --         1500     1280-1500
```

The following list describes some of the information that is displayed:

- The interface's property, its current value, and default value if not manually modified.
- If the property requires numeric values, the range of values is provided.
- Property permissions which would indicate if the property can be customized or not.
- The protocol to which the property definition applies, either IPv4 or IPv6.

Note - If any field value is unknown, such as when an interface does not support the property whose information is being requested, the value is displayed as a question mark (?).

Obtaining Information About IP Addresses

For information about IP addresses, use the `ipadm show-addr` *interface* command. If you do not specify an interface, then information about all of the IP addresses that are on the system is displayed.

```
$ ipadm show-addr
ADDROBJ           TYPE       STATE     ADDR
lo0/v4            static     ok        127.0.0.1/8
net0/v4           static     ok        198.51.100.3/24
tun0/v4           static     ok        203.0.113.1-->203.0.113.2
```

If you specify an interface with the command and the interface has multiple addresses, the following type of information is displayed:

```
$ ipadm show-addr net0
ADDROBJ           TYPE       STATE     ADDR
net0/v4           static     ok        198.51.100.3/24
net0/v4a          static     ok        192.0.2.1/24
net0/v4bc         static     ok        203.0.113.1
```

Note that if the address object pertains to a tunnel, then both the local and remote addresses are displayed in the output. See Chapter 5, “Administering IP Tunnels” in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

An address object that is displayed as *interface*/*?* indicates that the address was configured on the interface by an application that did not use `libipadm` APIs. Such applications are not under the control of the `ipadm` command.

Obtaining Information About IP Address Properties

For information about IP address properties, use the `ipadm show-addrprop` *addrobj* command. If you omit *addrobj*, information about is displayed. To list a single property for all of the IP addresses, specify just that property.

```
$ ipadm show-addrprop net1/v4
```

```
ADDROBJ   PROPERTY     PERM   CURRENT          PERSISTENT  DEFAULT          POSSIBLE
net1/v4   broadcast    r-     198.51.100.255   --          198.51.100.255   --
net1/v4   deprecated   rw     off              --          off              on,off
net1/v4   prefixlen    rw     24               24          24               1-30,32
net1/v4   private      rw     off              --          off              on,off
net1/v4   transmit     rw     on               --          on               on,off
net1/v4   zone         rw     global           --          global           --
```

Aside from the properties and their respective, permissions, current and default values, the command also displays persistent values. These values are reapplied whenever the system is booted.

Migrating From an IPv4 Network to an IPv6 Network

Note - Before you migrate from an IPv4 network to an IPv6 network, be aware that most migration plans involve running IPv4 and IPv6 together for an extended period of time, possibly indefinitely.

Prior to migrating from an IPv4 network to an IPv6 network, review the planning information in Chapter 2, “Planning for Using IPv6 Addresses” in *Planning for Network Deployment in Oracle Solaris 11.4* to determine whether you need to perform any additional tasks.

The basic steps for migrating from an IPv4 network to an IPv6 network involves removing all of the existing IPv4 DHCP and static IP addresses and reconfiguring new IPv6 addresses, as many as are required. If the new IPv6 interface is on the same link as a router that currently advertises an IPv6 prefix, the interface obtains the link prefix. See “Configuring an IPv6 Router” in *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer*.

EXAMPLE 8 Migrating IPv4 Addresses to IPv6 Addresses

The following examples show how to migrate your existing IPv4 addresses to IPv6 addresses. The process begins by deleting all of the existing IPv4 DHCP and static IP addresses.

```
$ ipadm show-addr net0/
ADDROBJ   TYPE     STATE   ADDR
lo0/v4    static   ok      127.0.0.1/8
net0/v4   static   ok      198.51.100.74/24
$ ipadm delete-addr net0/v4
```

For instructions, see “Modifying an IP Interface Configuration” on page 55.

Next, you create the new IPv6 address by using the `ipadm create addr` command with the appropriate options and arguments.

For example, you can create a link-local and `addrconf` IPv6 address as follows:

```
$ ipadm create-addr -T addrconf -p stateless=yes,stateful=yes net0/v6a
```

You can create a static IPv6 address without DHCPv6 and an `addrconf` address as follows:

```
$ ipadm create-addr -T addrconf -p stateless=no,stateful=no net0/v6a
$ ipadm create-addr -T static -a a::b/64 net0/v6b
```

To create a static IPv6 address, run the following command:

```
$ ipadm create-addr -T static -a a::b/64 net0/v6b
```

Display the new IPv6 configuration by using the `ipadm show-addr` command.

For additional IPv6 configuration steps, see “Configuring IPv6 Interfaces” on page 43.

♦ ♦ ♦ CHAPTER 4

Administering Naming and Directory Services on an Oracle Solaris System

This chapter describes how to configure naming services for an Oracle Solaris system and covers the following topics:

- "Overview of Naming and Directory Services Configuration"
- "About the `name-service/switch` SMF Service"
- "Configuring a DNS Client"
- "Importing Naming Services Configuration"
- "Resetting SMF Naming Services Configuration"

The tasks in this chapter assume that the system's IP interfaces have already been configured. You must have the appropriate rights profile to perform the procedures. See "Using Rights Profiles to Perform Network Configuration" on page 19.

Overview of Naming and Directory Services Configuration

A *naming service* performs lookups of stored information, such as host names and addresses, user names, passwords, access permissions, and so on. This information is made available so that users can log in to their systems, access resources, and be granted permissions. Naming service information can be stored in files, maps, or various forms of database files. These information repositories can be local to the system or located in a central network-based repository or database. Without a central naming service, each host would have to maintain its own copy of this information. If you centralize all data, administration becomes easier.

The following naming and directory services are supported in Oracle Solaris:

- Domain Name System (DNS) including Multicast DNS and DNS Service Discovery
- Lightweight Directory Access Protocol (LDAP)

About the name-service/switch SMF Service

The name-service/switch SMF service is a configurable selection service that enables you to specify which name service or source to use for each type of network information.

The service's configuration settings that used to be stored in the /etc/nsswitch.conf file are now property values of the SMF service. You must configure nsswitch settings through SMF rather than the file. The main commands to use are svccfg and svcadm. For example, to display current property settings of the name service, you would type:

```
$ svccfg -s name-service/switch listprop config
```

Configuring a DNS Client

DNS has two parts: a service that provides answers and a client that queries the service. This procedure assumes that the DNS server is already configured and operational.

The steps here apply only to configuring DNS on a client host. For server configuration instructions, see "Administering DNS" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

Note - LDAP configuration for naming services is not covered in this document. Like DNS, preparing LDAP involves both servers and clients and is fully described in *Working With Oracle Solaris 11.4 Directory and Naming Services: LDAP*.

▼ How to Configure a DNS Client in Oracle Solaris

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Prepare a list of domains to search and the IP addresses of the DNS name servers on your network.**

2. **Set properties in the SMF repository with the information from the previous step.**

   For example:

```
$ svccfg -s network/dns/client
svc:/network/dns/client> setprop config/search = astring: ("domain1" "domain2")
svc:/network/dns/client> setprop config/nameserver = net_address: (address1  address2)
```

```
svc:/network/dns/client> select network/dns/client:default
svc:/network/dns/client:default> refresh
svc:/network/dns/client:default> validate
```

*domain** refers to domain names such as `example.com`, `sales.my-example.com`, and so on. *address** are IP addresses of your DNS servers.

3. Specify the name service switch to use DNS.

```
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 system/name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch:default> validate
svc:/system/name-service/switch:default> quit
```

4. Start the services that are needed to run the DNS client.

```
$ svcadm enable dns/client
$ svcadm enable name-service/switch
```

5. Verify that the DNS configuration is properly set.

Various methods are available. The following list is only partial.

- Check for known domains.

  ```
  $ dig knownserver.example.com
  ```

- Start a lookup process of known domains.

  ```
  $ nslookup
  > www.oracle.com
  information displayed
  > exit
  $
  ```

- Open a browser and type a known URL to access a site.

Example 9 Setting Multiple DNS Options for a Client Simultaneously

The following example shows how to set multiple `/etc/resolv.conf` options simultaneously.

```
$ svccfg
svc:> select /network/dns/client
svc:/network/dns/client> setprop config/options = "ndots:2 retrans:3 retry:1"
svc:/network/dns/client> listprop config/options
config/options astring     ndots:2 retrans:3 retry:1
svc:/network/dns/client> exit
```

```
$ svcadm refresh dns/client
$ grep options /etc/resolv.conf
options ndots:2 retrans:3 retry:1
```

Enabling Multicast DNS

For Multicast DNS (mDNS) and DNS Service Discovery to function, mDNS must be deployed on all of the systems that will participate in mDNS. The mDNS service is used to advertise the availability of services that are provided on the system.

Prior to enabling mDNS, make sure that the software package is installed on your system. If necessary, install the package as follows:

```
$ pkg install pkg:/service/network/dns/mdns
```

Part of the process of enabling mDNS is to first update the name service switch information. To resolve local hosts, you must change the `config/host` property of the `name-service/switch` SMF service to include `mdns` as a source, as shown in the following example:

```
$ /usr/sbin/svccfg -s svc:/system/name-service/switch
svc:/system/name-service/switch> setprop config/host = astring: "files dns mdns"
svc:/system/name-service/switch> select system/name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch> quit
```

Enable the mDNS SMF service as follows:

```
$ svcadm enable svc:/network/dns/multicast:default
```

When you enable mDNS by using this method, your changes persist through upgrades and reboots. See the `svcadm`(8) man page.

Advertising Resources for DNS

You can use the `dns-sd` command to browse and discover services similar to how you use the `ping` or `traceroute` commands. The `dns-sd` command is primarily used interactively. The reason the command is used interactively is mainly because its command-line arguments and output format can change over time, which makes invoking the command from a shell script unpredictable and risky. Additionally, the asynchronous nature of DNS service discovery (DNS-SD) does not easily lend itself to script-oriented programming.

For examples, see "Advertising Resources for DNS" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*. See also the `dns-sd`(8) man page.

Importing Naming Services Configuration

You can use the `nscfg` command to transfer legacy file configuration for the `name-service` switch components to the SMF repository.

The following command imports any legacy file and then converts and pushes the configuration to SMF:

```
$ /usr/sbin/nscfg import -f FMRI
```

Using the `nscfg` command is the simplest way to populate the DNS configuration with information from a previously existing `resolv.conf` file. In the following example, the `nscfg` command reads the information in the `/etc/resolv.conf` file, converts it, then stores the information in the `svc:/network/dns/client` SMF service:

```
$ cp resolv.conf /etc/resolv.conf
$ /usr/sbin/nscfg import -f dns/client
$ svcadm enable dns/client
```

When you change a system's naming service, you must also modify the name service switch information accordingly and remove any stale information from the name service cache, as shown in the following example:

```
$ cp /etc/nsswitch.dns /etc/nsswitch.conf
$ /usr/sbin/nscfg import -f name-service/switch
$ svcadm refresh name-service/switch
$ svcadm refresh name-service/cache
```

See the `nscfg`(8) man page.

Resetting SMF Naming Services Configuration

Use the `unconfig` *FMRI* subcommand to reset the properties of an SMF naming service back to their default values. For example, to revert the changes to the SMF configuration in the previous section, you would type:

```
$ svcadm disable dns/client
$ /usr/sbin/nscfg unconfig dns/client
$ /usr/sbin/nscfg unconfig name-service/switch
$ svcadm refresh name-service/switch
$ svcadm refresh name-service/cache
```

♦ ♦ ♦ CHAPTER 5

Administering External Network Modifiers in Oracle Solaris

This chapter describes how to administer external network modifiers (ENMs) configuration in the current Oracle Solaris release. It contains the following topics:

Note - To administer external network modifiers and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 19.

About ENM Configuration

An external network modifier (ENM) is a type of network profile that manages applications that are responsible for creating network configuration that is external to the system's primary network configuration.

ENMs enable you to specify when applications or scripts should perform network or any other system configuration. ENMs can also be defined as services or applications that directly modify your network configuration when they are enabled or disabled. You can specify the conditions under which an ENM should be enabled or disabled, and you can also enable or disable an ENM manually. Multiple ENMs can potentially be active on the system at the same time.

You can create ENMs for several external applications and services. However, the obvious application of ENMs is on a virtual private network (VPN). After you install and configure VPN on your system, you can create an ENM that automatically activates and deactivates VPN under the conditions that you specify.

All ENMs have an `activation-mode` property. You set the `activation-mode` property when you create or modify an ENM with the `netcfg` command.

You can set the following values for this property:

- `manual` – Enabled and disabled by using the `netadm` command.
- `conditional-any` – Enabled and disabled by the system, based on the changes in the conditions of the ENM.
- `conditional-all` – Enabled and disabled by the system, based on changes in the conditions of the ENM.

Changes in the network environment cause the system to reevaluate conditions for the ENM. These changes include plugging or unplugging an Ethernet cable, obtaining or losing a DHCP lease, detecting a new wireless network, and so on.

Enabling and Disabling ENMs

You use the `netadm` command to enable and disable ENMs. The basic command syntax is as follows:

```
$ netadm enable enm-name
```

For example, you can enable an ENM named `test-enm1` as follows:

```
$ netadm enable test-enm1
```

ENMs can have a `manual` or `conditional` activation mode. If you set the `activation-mode` property to `conditional`, the system enables or disables the ENM based on the specified conditions. If you set the activation mode property to `manual`, you can enable or disable the ENM by using the `netadm` command. There are no constraints on ENM activation. Zero or more ENMs can be active on a system at any given time. Enabling or disabling an ENM has no effect on other currently active ENMs.

For background information about ENM activation modes, see "About ENM Configuration" on page 69.

EXAMPLE 10 Enabling ENMs

In the following example, an ENM named `test-enm1` is enabled.

```
$ netadm enable enm test-enm1
Enabling enm 'test-enm1'
```

EXAMPLE 11 Disabling ENMs

In the following example, an ENM named `test-enm2` is disabled.

```
$ netadm disable test-enm2
Disabling enm 'test-enm2'
```

Configuring ENMs

You can configure ENMs by using the `netcfg` command in the following three ways:

- Interactively
- Command-line mode
- Command-file mode

Configuring ENMs Interactively

When used interactively, the concept of a *scope* is used for the `netcfg` command. When you use the command interactively, the scope that you are in at any given time depends on the particular task that you are performing. When you type the `netcfg` command by itself in a terminal window, as shown in the following example, a prompt is displayed at the *global scope*:

```
$ netcfg
netcfg>
```

To create or select an ENM, first initiate the `netcfg` interactive session.

From the global scope prompt, you can use the `select` or `create` subcommands to view, modify, or create an ENM. After you have created or selected an ENM, the syntax of the `netcfg` interactive prompt looks similar to the following example:

`netcfg:enm:`*enm-name*`>`

Use the `netcfg` command in the interactive mode to perform the following tasks:

- Create an ENM.
- Select and modify an ENM.
- Verify that all of the required information about an ENM is valid.
- Commit the changes for a new ENM.
- Cancel the current ENM configuration without committing any changes to persistent storage.

- Revert the changes that you made for an ENM.

Selecting or creating an ENM while in the interactive mode results in a command prompt that is displayed at the *profile scope*, as shown in the following example:

```
netcfg> select test-enm1
netcfg:enm:test-enm1>
```

At any given scope, the command prompt indicates the currently selected ENM. You can commit any changes that you make to the ENM in this scope, which means the changes are saved to persistent storage. Changes are also implicitly committed upon exiting the scope. If you do not want to commit the changes that you made, you can revert to the previously committed state for the ENM, which reverts any changes that you made to the ENM at that level. The `revert` and `cancel` subcommands work similarly.

Note - The `walkprop` subcommand is used to display each property that is associated with an ENM. This command is meaningful only when used interactively.

How to Create an ENM Interactively

ENMs enable you to specify when applications or scripts such as a VPN application should perform network configuration.

Note - The system does not automatically recognize an application for which you might create an ENM. These applications must first be installed and then configured on the system before creating an ENM for them by using the `netcfg` command.

See the `netcfg`(8) man page.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Initiate the `netcfg` interactive session.**

```
$ netcfg
netcfg>
```

2. **Create the ENM.**

```
netcfg> create test-enm
Created enm 'test-enm'.  Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]>
fmri> svc:/application/test-enm:default
```

```
start>
stop>
netcfg:enm:test-enm>
```

Creating the ENM automatically moves you to the profile scope for the ENM and walks each of its properties.

In this example, the following properties are specified for the `test-enm` ENM:

- The `activation-mode` property, which is set to manual, is accepted by pressing Return. Because this value is set to `manual`, the `conditions` property is not made available for setting.
- The `fmri` property is set to `svc:/application/test-enm:default`.
- The `start` and `stop` properties are not set for this ENM.

3. (Optional) Display the profile configuration.

```
netcfg:enm:test-enm> list
enm:test-enm
     activation-mode   manual
     enabled           false
     fmri              "svc:/application/test-enm:default"
```

4. Verify that the profile configuration is correct.

```
netcfg:enm:test-enm> verify
All properties verified
```

The `verify` subcommand verifies the configuration and notifies you if any required values are missing.

5. Save the ENM by committing the changes.

```
netcfg:enm:test-enm> commit
Committed changes
netcfg>
```

The `commit` subcommand verifies and saves the configuration.

Alternatively, you can use the `end` subcommand to end the session, which also saves the profile configuration.

```
netcfg:enm:test-enm> end
Committed changes
```

6. Exit the interactive session.

```
netcfg> exit
```

Configuring ENMs in Command-Line Mode

In command-line mode, any `netcfg` subcommand that affects a selected ENM or property must be performed within the particular scope in which the selected ENM or property exists. Selecting an ENM in command-line mode is the same as selecting an ENM in the interactive mode. The only difference is that in command-line mode all of the subcommands are written on the command-line.

For example, to list an ENM's configuration, you would first select it and then use the `list` subcommand to display the properties for that ENM:

```
$ netcfg "select myenm; list"
enm:myenm
        activation-mode         manual
        enabled                 false
        start                   "/var/tmp/start_myenm"
```

Refer to the following guidelines when using the `netcfg` command in command-line mode:

- Separate each subcommand by a semi-colon.
- Specify the `select` subcommand from the global scope to move to the profile scope of the ENM.
- Specify the `list` subcommand from the profile scope to list the properties that are within that scope.
- Use straight quotation marks to prevent the shell from interpreting the semi-colons.

Note - In command-line mode, you must type the complete command on a single line. Changes that you make to a selected profile by using the `netcfg` command in command-line mode are committed to persistent storage as soon as the command is executed.

You can use any of the `netcfg` subcommands in command-line mode, except for the `walkprop` subcommand.

Working in the `netcfg` Command-File Mode

In command-file mode, profile configuration information and commands are extracted from a file. The commands in the file are same as those that are used in the interactive mode and by specifying the `export` subcommand. The `export` subcommand when specified with the `-f` option, produces the file.

For example, the following command exports the current profile configuration to a file:

```
$ netcfg export -f /tmp/nwam.config
```

To import a profile configuration from a file, type the following command:

```
$ netcfg -f /tmp/nwam.config
```

You can also use the `export` subcommand interactively. For more information, see "Exporting an ENM Configuration" on page 84.

Administering ENMs

This section describes different tasks for managing ENMs.

Setting Property Values for ENMs

You set or change property values for ENMs by using the `set` subcommand. You can use this subcommand interactively or in command-line mode. If you set or change property values in command-line mode, the change is immediately committed to persistent storage.

The syntax that you use for the `set` subcommand is as follows:

`netcfg:enm:`*enn-name*`> set` *prop-name*`=`*value1*`[,`*value2*`,...]`

▼ How to Set ENM Property Values Interactively

The following procedure describes how to set property values for an ENM interactively. When setting property values interactively, you must first select an ENM from the global scope, which moves the interactive session into that ENM's profile scope. The selected ENM is then loaded into memory from persistent storage. In this scope, you can then modify the properties of the ENM.

For example purposes only, the following procedure shows how to set the `start` property of the `test-enm` ENM interactively.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Initiate an `netcfg` interactive session.**

```
$ netcfg
netcfg>
```

2. **Select the ENM that you want to modify.**

```
netcfg> select test-enm
netcfg:enm:test-enm>
```

3. **Set the property value.**

In the following example, the `start` property is set:

```
netcfg:enm:test-enm> set start = /path/to/start-script
```

4. **Display the configuration information.**

```
netcfg:enm:test-enm> list
enm:myenm
        activation-mode         manual
        enabled                 false
        start                   "/path/to/start-script"
```

5. **End the session.**

```
netcfg:enm:test-enm> end
Committed changes
netcfg>
```

The `end` subcommand saves and moves the session to the global scope.

6. **Exit the interactive session.**

```
netcfg> exit
```

Example 12 Setting Property Values for an ENM in Command-Line Mode

You can also set the `start` property for an ENM in command-line mode as follows:

```
$ netcfg "select test-enm; set start =  /path/to/start-script"
```

The command-line mode is best suited for when you just need to perform a simple action. However, you can also use the command-line mode to perform more complex actions by carefully specifying the appropriate subcommands on the command line. As with the interactive example, in command-line mode you must also first select the ENM to move into that ENM's profile scope. You can then specify the `set` subcommand to set the individual property values.

When using the command-line mode, you can set multiple values for a given property at the same time. When setting multiple values on the command line, you must separate each value by a comma (,). If individual values for a specified property also contain a comma, the comma that is part of the property value must be preceded by a backslash (\). Commas within properties that only have a single value are not interpreted as delimiters and therefore do not need to be preceded by a backslash.

Obtaining Information About ENM Configuration

Use the `list` subcommand to list all ENMs, property-value pairs, and resources that exist at the current or specified scope, either interactively or in command-line mode.

In the global scope, the `list` subcommand lists all of the ENMs on a system, as shown in the following example:

```
$ netcfg list
ENMs:
              test-enm
              myenm
```

Note - In the interactive mode, using the `list` subcommand in the global scope displays the same information.

Note that the `list` subcommand does not list the state of each ENM. To display information about ENMs and their states, use the `netadm list` command. For more information, see "Displaying Information About ENMs" on page 82.

Listing Property Values for an Individual ENM

The `list` subcommand in the profile scope lists all of the property values for an ENM. The syntax is as follows:

```
netcfg> list [enm-name]
```

▼ How to List All of the Property Values for an ENM Interactively

The following procedure describes how to list all of the property values for an ENM by using the `list` subcommand interactively. The example in the following procedure shows how to

list the configuration information for an ENM named `myenm`.The values that are listed for each ENM vary.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Initiate the `netcfg` interactive session.**

```
$ netcfg
netcfg>
```

2. **List the configuration for the ENM by using one of the following methods:**

   - **List the configuration from the global scope:**

```
netcfg> list myenm
enm:myenm
        activation-mode         manual
        enabled                 false
        start                   "/path/to/start-script"
```

   - **List the configuration from the ENM scope:**

```
netcfg> select myenm
netcfg:enm:myenm> list
enm:myenm
        activation-mode         manual
        enabled                 false
        start                   "/path/to/start-script"
```

3. **Exit the interactive session.**

```
netcfg:enm:myenm> exit
```

Example 13 Listing Property Values for an ENM in Command-Line Mode

You can also list property values for an ENM by using the command-line mode. The output is identical, regardless of which mode you use.

For example, you can list the properties of the ENM named `myenm` from the global scope as follows:

```
$ netcfg "list myenm"
```

You can list the properties for an ENM from the ENM's profile scope as follows:

```
$ netcfg "select myenm; list"
```

Obtaining a Specific Property Value for an ENM

Use the `get` subcommand to obtain a specific property value for an ENM. You can use this subcommand either interactively or in command-line mode. The command syntax is as follows:

```
netcfg:enm:enm-name> get [ -V ] prop-name
```

▼ How to Obtain a Specific Property Value for an ENM

The following procedure describes how to obtain a specific property value for an ENM by using the `get` subcommand interactively. The example in the following procedure shows how to obtain the `start` property for an ENM named `myenm`.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Initiate the `netcfg` interactive session.**

```
$ netcfg
netcfg>
```

2. **Select the ENM.**

```
netcfg> select myenm
```

3. **Obtain the specific property value by using one of the following commands:**

   - **Use the `get` subcommand to display the property name and the property value as follows:**

```
netcfg:enm:myenm> get start
            start                   "/path/to/start-script"
```

   - **If you want to obtain only the property value, and not the property name, use the `-V` option with the `get` subcommand:**

```
netcfg:enm:myenm> get -V start
/path/to/start-script
```

4. **Exit the interactive session.**

```
netcfg:enm:myenm> exit
```

Example 14 Obtaining a Specific Property Value for an ENM in Command-Line Mode

You can also obtain a specific property value for an ENM in command-line mode. The output is identical, regardless of which mode you use.

For example, you can obtain the value of the `start` property for an ENM (`myenm`) in command-line mode as follows:

```
$ netcfg "select myenm; get start"
            start                   "/path/to/start-script"
```

The following example shows how to use the `get` subcommand with the `-V` option to obtain a specific property value. This method is useful for scripts, where the property name does not need to be parsed.

```
$ netcfg "select myenm; get -V start"
/path/to/start-script
```

Setting Property Values for an ENM by Using the `walkprop` Subcommand

You use the `walkprop` subcommand to interactively view and change individual property values for an ENM. After initiating the interactive session, type the `walkprop` subcommand to display the name and current value for each of the properties of an ENM, one property at a time. As you view the various properties, you can set or change the current or default value, as desired.

Note - The `walkprop` subcommand is meant to be used in the interactive mode *only*.

▼ How to Use the `walkprop` Subcommand to View and Set Property Values for an ENM

The following procedure describes how to use the `walkprop` subcommand to interactively view and change property values for a given ENM. As shown in the following examples, when you use the `walkprop` subcommand to set the properties of an ENM, you do not have to use the `set` subcommand.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Initiate the `netcfg` interactive session.**

```
$ netcfg
netcfg>
```

2. **Select the ENM for which you want to view and change properties.**

   In the following example, an ENM named `test-enm` is selected:

   ```
   netcfg> select test-enm
   netcfg:enm:test-enm>
   ```

3. **Type the `walkprop` subcommand to start the walk.**

   In the following example, after the `walkprop` subcommand is issued, the first property that is displayed is the `activation-mode` property. Note the default value of the property, which is currently set to `manual` (shown in parentheses).

   ```
   netcfg:enm:test-enm> walkprop
   activation-mode (manual) [manual|conditional-any|conditional-all]>
   ```

4. **To change a property's value, type the new value at interactive prompt, then press Return.**

   For example, you would change the `activation-mode` property for an ENM from `manual` to `conditional-all` as follows:

   ```
   netcfg:enm:test-enm> walkprop
   activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
   ```

   Pressing Return saves the current setting and displays or "walks" the next property.

5. **Repeat the walk process until all of the properties for the profile have been displayed, making modifications as needed, per the instructions in Step 4.**

   ```
   netcfg:enm:test-enm> walkprop
   activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
   conditions> advertised-domain is example.com
   fmri ("svc:/application/test-enm:default")>
   start>
   stop>
   ```

   Pressing Return without making any changes retains the existing default value for the property and then advances the walk to the next property.

   Note - Only the relevant properties for a given ENM are displayed, as described in "Configuring ENMs Interactively" on page 71.

6. **List the current default property values for the ENM as follows:**

```
netcfg:enm:test-enm> list
enm:test-enm
        activation-mode         conditional-all
        conditions              "advertised-domain is example.com"
        enabled                 false
        fmri                    "svc:/application/test-enm:default"
```

Note that in the previous output, the `activation-mode` property is now set to `conditional-all`.

7. **Exit the interactive session to commit the changes.**

```
netcfg:enm:test-enm> exit
Committed changes
```

Displaying Information About ENMs

Use the `netadm` command with the `list` subcommand to display information about any or all of the ENMs on a system, including the current state of each ENM.

The syntax for the `list` subcommand is as follows:

```
$ netadm list [ enm-name ]
```

For example, you can display all of the ENMs on a system and their current state as follows:

```
$ netadm list
ENM             STATE
myenm           online
test-enm        disabled
```

EXAMPLE 15 Displaying the Current State of a Specified ENM

The following example lists the current state of the ENM named `myenm`.

```
$ netadm list myenm
ENM             STATE
myenm           online
```

EXAMPLE 16 Displaying Auxiliary State Values of ENMs

The auxiliary state of an ENM provides an explanation about why a given ENM is online or offline (enabled or disabled). To list auxiliary state values, use the `-x` option with the `list` subcommand as follows:

```
$ netadm list -x
ENM             STATE          AUXILIARY STATE
myenm           online         active
test-enm        disabled       disabled by administrator
```

Removing ENMs From the System

Use the `destroy` subcommand to remove ENMs from the system. You must first disable the ENM, and then remove it.

The syntax for the `destroy` subcommand is as follows:

```
netcfg> destroy [ -a | enm-name ]
```

The `-a` option removes all of the ENMs from the system, except for any currently active ENMs.

▼ How to Remove an ENM Interactively

The following procedure describes how to remove an ENM interactively. For example purposes only, this procedure shows how to remove an ENM named `test-enm`.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Initiate the `netcfg` interactive session.**

   ```
   $ netcfg
   netcfg>
   ```

2. **Remove the profile or configuration object.**

   For example, you can remove an ENM named `test-enm` as follows:

   ```
   netcfg> destroy test-enm
   netcfg>
   ```

3. **Exit the interactive session.**

   ```
   netcfg> exit
   ```

Example 17 Removing an ENM in the Command-Line Mode

You can also remove an ENM by using the command-line mode as follows:

```
$ netcfg "destroy test-enm"
```

Exporting an ENM Configuration

Use the `export` subcommand to save an ENM configuration. Exporting an ENM can be useful if you are responsible for maintaining multiple servers that require identical network configurations. You can use the `export` subcommand either interactively or in command-line mode. When an ENM is exported, the output is displayed as a series of subcommands that the `netcfg` command is capable of interpreting. These subcommands are similar to commands that you can type in the interactive or command-line mode.

The syntax for the `export` subcommand is as follows:

```
netcfg> export [ -d ] [ -f output-file ] [ enm-name ]
```

Note - The `-d` and `-f` options of the `export` subcommand can be used independently of each other. The `-f` option prints the current configuration at the current or specified scope to a specified file. The `-d` option adds the `destroy -a` command as the first line of output.

EXAMPLE 18 Exporting an ENM Configuration Interactively

The following example shows how to display an ENM configuration on-screen by using the `export` subcommand interactively.

```
$ netcfg
netcfg> export
create "myenm"
set activation-mode=manual
set start="/path/to/start-script"
end
create "myenm"
set activation-mode=conditional-all
set conditions="advertised-domain is example.com"
set fmri="svc:/application/test-enm:default"
end
```

Using command-line mode, you can type the following command:

```
$ netcfg export
```

You can use the `-d` option with the `export` subcommand to add the `destroy -a` command as the first line of the `netcfg export` output, as shown in the following example, which has been truncated.

```
$ netcfg> export -d
destroy -a
create "myenm"
set activation-mode=manual
set start="/path/to/start-script"
...
```

In command-line mode, you can type the following command:

```
$ netcfg export -d
```

EXAMPLE 19 Exporting an ENM Configuration to a File

In the following examples, the configuration information for the `test-enm` ENM is written to a file by using the `export` subcommand with the `-f` option. The `-f` option writes the output to a new file named `myconf`, while the `-d` option adds the `destroy -a` command as the first line of the `netcfg export` output.

You can export the profile configuration to a file interactively as follows:

```
$ netcfg
netcfg> export -d -f myconf
```

You can perform the same task in command-line mode as follows:

```
$ netcfg export -d -f myconf
```

The following truncated example shows how to display the profile configuration:

```
$ cat myconf
destroy -a
create "test-enm"
.
.
.
```

Restoring an Exported ENM Configuration

You can import an ENM configuration that was generated with the `export` subcommand back into the system. The exported file has a series of subcommands that the `netcfg` command is capable of interpreting. The `export` subcommand is useful if you need to restore an ENM configuration or copy an ENM configuration to another system. Note that you can only restore an ENM configuration by using the `netcfg` command-line mode.

Use the following command to restore an exported configuration:

```
$ netcfg -f file
```

For example, you can restore the file that was exported in Example 19, “Exporting an ENM Configuration to a File,” on page 85 as follows:

```
$ netcfg -f myconf
Configuration read.
```

♦ ♦ ♦ CHAPTER 6

Administering Wireless Networks in Oracle Solaris

This chapter describes how to administer wireless networks in the current Oracle Solaris release and contains the following topics:

Oracle Solaris does not include features for configuring WiFi servers or access points.

Note - To administer wireless networks and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 19.

Connecting to and Monitoring a WiFi Network

The IEEE 802.11 specifications define wireless communications for local area networks. These specifications and the networks they describe are referred to collectively as *WiFi*, a term that is trademarked by the Wi-Fi Alliance trade group. WiFi networks are reasonably easy to configure by both providers and prospective clients.

▼ How to Connect to a WiFi Network

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Display the physical attributes of the datalinks that are on the system.**

```
$ dladm show-phys
```

```
LINK            MEDIA                STATE    SPEED  DUPLEX    DEVICE
net0            Ethernet             up       1500   full      e1000g0
net1            WiFi               down       1500   full      ath0
```

The output indicates that two links are available. `net0` over the device `e1000g0`, which supports wired communications and `net1` over the device `ath0`, which enables you to connect to a wireless network.

2. **Configure the WiFi interface.**

   a. **Create the interface that supports WiFi.**

   ```
   $ ipadm create-ip net1
   ```

   b. **Verify that the link has been plumbed.**

   ```
   $ ipadm show-if
   IFNAME      CLASS        STATE     ACTIVE     OVER
   lo0         loopback     ok        yes        --
   net1        ip           ok        yes        --
   ```

3. **Check for available networks.**

```
$ dladm scan-wifi
LINK       ESSID        BSSID/IBSSID       SEC     STRENGTH   MODE   SPEED
net1       ofc          00:0e:38:49:01:d0  none     good       g      54Mb
net1       home         00:0e:38:49:02:f0  none     very weak  g      54Mb
net1       linksys      00:0d:ed:a5:47:e0  none     very good  g      54Mb
```

The `scan-wifi` command displays information about the available WiFi networks at the current location. The output includes the following information:

`LINK` — Link name to be used in the WiFi connection.

`ESSID` — Extended Service Set ID. The `ESSID` is the name of the WiFi network, which can be randomly named by the administrator of the specific wireless network.

`BSSID/IBSSID` — Basic Service Set ID (BSSID), which is a unique identifier for a particular ESSID. The BSSID is the 48-bit MAC address of the nearby access point that serves the network with a particular ESSID.

`SEC` — Type of security that is required to access the wireless network. The values are `none` and `WPA`. For more information, see “Establishing Secure WiFi Communications” on page 96.

`STRENGTH`	Refers to the strength of the radio signals from the WiFi networks that are available at your location.
`MODE`	Version of the 802.11 protocol that is run by the network. The modes are a, b, and g, or any combination of these modes.
`SPEED`	Refers to the speed (in megabits per second) of the particular network.

4. **Connect to a WiFi network.**

```
$ dladm connect-wifi [-e ESSID]
```

If a connection is established, a Known WLAN is created. See "Administering Known WLANs" on page 92 for more details about Known WLANs.

Note - Successfully connecting to a WiFi network creates a Known WLAN. See "Administering Known WLANs" on page 92 for more details about Known WLANs.

For more information about using the `dladm connect-wifi` command, see "Establishing Secure WiFi Communications" on page 96. See also the `dladm`(8) man page.

5. **Check the status of the WiFi network to which the system is connected.**

```
$ dladm show-wifi
LINK        STATUS        ESSID       SEC     STRENGTH   MODE   SPEED
net1        connected     ofc         none    very good   g      36Mb
```

The previous output indicates that the system is connected to the `ofc` network. The `scan-wifi` output from Step 3Step 4 of this procedure indicated that `ofc` had the strongest signal of the available networks. The `dladm connect-wifi` command automatically chooses the WiFi network with the strongest signal, unless you explicitly specify a different wireless network.

6. **Configure an IP address for the interface by using either of the following methods:**

- **Obtain an IP address from a DHCP server.**

```
$ ipadm create-addr -T dhcp interface
```

- **Configure a static IP address.**

```
$ ipadm create-addr -a address interface
```

Use this option if you have a dedicated IP address for the system.

7. **Access the Internet through the WiFi network.**

8. **To end the session, use either the `dladm disconnect-wifi` command or shut down the system.**

Example 20 Connecting to a Specific WiFi Network

The following example combines the different steps that you can take to connect your Oracle Solaris system to a wireless network. The example also shows how you can force the system to connect to a specific and preferred wireless network instead of allowing the OS to randomly select the wireless network. In the following example, assume that the static IP address `203.0.113.3/24` is assigned for use on the system.

```
$ dladm show-phys
LINK            MEDIA               STATE    SPEED  DUPLEX    DEVICE
net0            Ethernet              up       1500   full      e1000g0
net1            Wifi                down       1500   full      ath0

$ ipadm create-ip net1
$ ipadm show-if net1
IFNAME      CLASS        STATE     ACTIVE     OVER
net1        ip           ok        yes        --

$ dladm scan-wifi
LINK       ESSID        BSSID/IBSSID       SEC     STRENGTH   MODE   SPEED
net1       wifi-a       00:0e:38:49:01:d0  none     weak       g      54Mb
net1       wifi-b       00:0e:38:49:02:f0  none     very weak  g      54Mb
net1       ofc          00:0d:ed:a5:47:e0  none     very good  g      54Mb
net1       mynet        00:40:96:2a:56:b5  none     good       b      11Mb

$ dladm connect-wifi -e mynet
$ dladm show-wifi
LINK       STATUS        ESSID         SEC     STRENGTH   MODE   SPEED
net1       connected     mynet         none    good       g      11Mb

$ ipadm create-addr -a 203.0.113.3/24 net0
ipadm: net1/v4
$ ipadm show-addr net0
ADDROBJ           TYPE     STATE        ADDR
net1/v4           static   ok           203.0.113.3/24
```

Launch a browser or another application to commence your work over the WiFi network.

Terminate the session but leave the system running.

```
$ dladm disconnect-wifi
$ dladm show-wifi
LINK       STATUS        ESSID         SEC     STRENGTH   MODE   SPEED
```

```
net1       disconnected   --            --       --       --       --
```

The output of the `show-wifi` command verifies that you have disconnected the `net1` link from the WiFi network.

How to Monitor a WiFi Link

The following procedure describes how to monitor the status of a WiFi link through standard networking tools and also how to change a selected link property.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 19.

1. **Connect to a WiFi network, as described in "How to Connect to a WiFi Network" on page 87.**

2. **View the properties of the system's datalinks.**

   ```
   $ dladm show-linkprop link
   ```

3. **Set a fixed speed for the link.**

   Caution - Oracle Solaris automatically selects the optimal speed for a WiFi connection. Modifying the initial speed of the link might diminish performance or prevent the establishment of certain WiFi connections.

   You can modify the link speed to one of the possible values that are listed in the output of the `show-linkprop` subcommand, as shown in the following example:

   ```
   $ dladm set-linkprop -p speed=value link
   ```

4. **Check the packet flow over the link.**

   ```
   $ netstat -I net1 -i 5
   input   net1      output       input  (Total)    output
   packets errs  packets errs  colls  packets errs  packets errs  colls
   317     0     106     0     0      2905    0     571     0     0
   14      0     0       0     0      20      0     0       0     0
   7       0     0       0     0      16      0     1       0     0
   5       0     0       0     0      9       0     0       0     0
   304     0     10      0     0      631     0     316     0     0
   338     0     9       0     0      722     0     381     0     0
   294     0     7       0     0      670     0     371     0     0
   306     0     5       0     0      649     0     338     0     0
   ```

```
289     0     5       0     0      597     0     301     0     0
```

Example 21 Setting the Speed of a Link

The following example shows how to set the speed of a link after you have connected to a WiFi network.

```
$ dladm show-linkprop -p speed net1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net1     speed           r-   25           0            0
 1,2,5,6,9,11,12,18,24,36,48,54
$ dladm set-linkprop -p speed=36 net1

$ dladm show-linkprop -p speed net1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net1     speed           r-   36           0            0
 1,2,5,6,9,11,12,18,24,36,48,54
```

Administering Known WLANs

This selection discusses different topics about wide local area networks.

About Known WLANs

You use *Known WLANs* to manage wireless networks that are known to the system. Usually, these are wireless networks that the system has connected to previously. Additionally, you can also manually create Known WLANs. The system then maintains a list of these known wireless networks. A Known WLAN stores the priority of that wireless network, as well as the any password or key that might be associated with that wireless network. You create this key by using the `dladm create-secobj` subcommand.

Creating and Administering Known WLANs

In this Oracle Solaris release, you configure and administer known WLANs by using the `dladm` command. See the `dladm`(8) man page.

Creating a Known WLAN

Use the following command syntax:

```
$ dladm create-wlan wlan-name
```

Whenever you connect to a WiFi network from your notebook PC, a Known WLAN is automatically created as soon as the `dladm connect-wifi` command succeeds. Properties for the Known WLAN are also set at that time.

To connect to a wireless network in the Known WLANs list, use the `dladm connect-wifi` command with the `-w` option as follows:

```
$ dladm connect-wifi -w
```

Running this command connects to the available wireless network in the Known WLAN list that has the highest priority. If the wireless network requires a key, you do not have to provide it. The key information from the Known WLAN is used to connect to that wireless network.

For more information about manually connecting to a WiFi network and switching WiFi networks, see "How to Connect to a WiFi Network" on page 87.

Displaying Configuration Information for a Known WLAN

Use the following command syntax:

```
$ dladm show-wlan wlan-name
```

To include BSSID information in the output, include the `-o all` option with the `show-wlan` subcommand.

```
$ dladm show-wlan -o all wlan-name
```

EXAMPLE 22 Creating and Displaying Known WLANs

The following example shows how to create a Known WLAN named `home-wifi`.

```
$ dladm create-wlan home-wifi
```

Display configuration information about the Known WLAN as follows:

```
$ dladm show-wlan
ESSID               PRIORITY  SECURITY-MODE   KEY
home-wifi           0         --              --
office-wifi         2         --              --
```

Display BSSID information for the Known WLAN as follows:

```
$ dladm show-wlan -o all
```

```
ESSID               PRIORITY  BSSIDS                  SECURITY-MODE   KEY
home-wifi           0         --                      --              --
office-wifi         2         --                      --              --
```

Setting Properties for a Known WLAN

Use the following command syntax:

```
$ dladm set-wlan -p property=value1[,...] wlan-name
```

Note that property values are comma separated.

EXAMPLE 23 Setting the Priority for a Known WLAN

By default, Known WLANs are assigned the highest priority `0` when created. The following example shows how to change the priority for a Known WLAN named `home-wifi` to `1`. For more information, see "About Known WLANs" on page 92.

```
$ dladm set-wlan -p priority=1 home-wifi
```

EXAMPLE 24 Setting the Security Mode and Adding a WPA Security Key for a Known WLAN

The following example shows how to change the security mode for a Known WLAN (`home-wifi`) to WPA and also how to add a security key. Note that you must create the key name prior to setting this property value for a Known WLAN.

Type the following command to check that the key name has been created:

```
$ dladm show-secobj
OBJECT                  CLASS
home-key                  wpa
```

Set the `security-mode` and `key` properties as follows:

```
$ dladm set-wlan -p security-mode=wpa,key=home-key home-wifi
$ dladm show-wlan -o all
ESSID               PRIORITY  BSSIDS                  SECURITY-MODE   KEY
home-wifi           1         a:b:c:d:e:f,f:e:d:c:b:a wpa             home-key
office-wifi         2         -                       --              --
```

EXAMPLE 25 Setting Multiple BSSIDs for a Known WLAN

The following example shows how to set multiple BSSIDs for a Known WLAN, `home-wifi`. Note that each BSSID is separated by a comma.

```
$ dladm set-wlan -p bssids=a:b:c:d:e:f,f:e:d:c:b:a home-wifi
$ dladm show-wlan -o all
ESSID               PRIORITY  BSSIDS                  SECURITY-MODE   KEY
home-wifi           1         a:b:c:d:e:f,f:e:d:c:b:a wpa             home-key
office-wifi.....    2..       --..                    --              --
```

EXAMPLE 26 Resetting Property Values for a Known WLAN

You can reset property values for a Known WLAN by using the `reset-wlan` subcommand. This command resets the specified property value back to its original, default value. The following example shows how to reset the `security-mode` and `key` property values for the `home-wifi` WLAN back to the default values.

```
$ dladm show-wlan
ESSID               PRIORITY  SECURITY-MODE   KEY
home-wifi           1         wpa             home-key
office-wifi.....    2..       --..            --

$ dladm reset-wlan -p security-mode,key home-wifi
$ dladm show-wlan
ESSID               PRIORITY  SECURITY-MODE   KEY
home-wifi           1         --              --
office-wifi.....    2..       --..            --
```

Deleting a Known WLAN

Use the following command syntax:

```
$ dladm delete-wlan wlan-name
```

For example, you would delete a Known WLAN named `home-wifi` as follows:

```
$ dladm delete-wlan home-wifi
```

EXAMPLE 27 Deleting a Known WLAN by Using the `dladm disconnect-wifi` Command

You can also delete a known WLAN by using `disconnect-wifi` subcommand with the `-d` option, as shown in the following example.

```
$ dladm disconnect-wifi -d link
$ dladm show-wifi
LINK       STATUS        ESSID         SEC     STRENGTH   MODE   SPEED
net1       disconnected   --            --       --       --       --
```

where *link* represents the interface that was used to configure the Known WLAN.

Managing Known WLAN Behavior at Boot Time

When you boot your system, the `nwamd` daemon automatically connects to the available wireless network in the Known WLAN list with the highest priority. The daemon also monitors the network, and if it detects a dropped connection, the daemon reconnects to the network. This feature is controlled by the `auto-connect` link property for each link that is on a system. The resulting behavior is that the `nwamd` daemon does not automatically connect to a wireless network. To connect to a wireless network, you must do so manually by using the `dladm connect-wifi` command. You can specify the `-w` option with the command to connect to the available wireless network in the Known WLAN list with the highest priority. When a connection is established, the `nwamd` daemon monitors that connection and will reconnect if a disconnect is detected.

By default, the `auto-connect` property is set to `on`. However, you can turn this feature off by using the `set-linkprop` subcommand of `dladm` command as follows:

```
$ dladm set-linkprop -p auto-connect=off link
```

Establishing Secure WiFi Communications

Radio wave technology makes WiFi networks readily available and often freely accessible to users. As a result, connecting to a WiFi network can be an insecure undertaking.

The following types of WiFi connections are more secure:

- Connecting to a private, restricted-access WiFi network.

  Private networks, such as internal networks that are established by corporations or universities, restrict network access to users who can provide the correct security challenge. Potential users must supply a key during the connection sequence or log in to the network through a secure VPN application.
- Encrypting your connection to a WiFi network.

  You can encrypt communications between your system and a WiFi network by using a secure key. Your access point to the WiFi network must be a router that is in your home or office with a secure key-generating feature. Your system and the router establish and then share the key before creating the secure connection.

The `dladm` command can use a WiFi Protected Access (WPA) key for encrypting connections through an access point. The WPA protocol is defined in the IEEE 802.11i specifications for wireless connections. Oracle Solaris supports version 2 of the WPA standard. See the `dladm`(8) man page.

♦ ♦ ♦ APPENDIX A

Datalink Properties

Name Changes of Datalink Properties

The following table lists the original `dladm` property name and the new, corresponding property name.

TABLE 2 Property Name Changes for the `dladm` Command

Original `dladm` Property Name	New `dladm` Property Name	Description
`autoconnect`	`auto-connect`	Name change only
`powermode`	`power-mode`	Name change only
`linkmode`	`link-mode`	Name change only
`adv_autoneg_cap`	`auto-negotiation`	Name change only
`flowctrl`	`flow-control`	Name change only

`en_10gfdx_cap`
`en1000fdx_cap`
`en1000hdx_cap`
`en_100fdx_cap`
`en_100hdx_cap`
`en10_fdx_cap`
`en10hdx_cap`
	`speed-duplex`	The following are examples of the values for enabled speed-duplex combinations that you can specify
		`10g-f` (10 gigabit full-duplex)
		`1g-f` (1 gigabit full-duplex)
		`1g-h` (1 gigabit half-duplex)
		`100m-f` (100megabit full-duplex)
		`100m-h` (100megabit half-duplex)
		`10m-f` (10 megabit full-duplex)
		`10m-h` (10 megabit half-duplex)
		Current values are advertised speed-duplex combinations, for example, the equivalent of the `adv_`*speed-duplex*`_cap` values.

`maxbw`	`max-bw`	Name change only Note that this property name change also applies to administering flows with the `flowadm` command.
`bwshare`	`bw-share`	Name change only

Original `dladm` Property Name	New `dladm` Property Name	Description
`rxfanout`	`rx-fanout`	Name change only
`tagmode`	`tag-mode`	Name change only
`hoplimit`	`hop-limit`	Name change only
`encaplimit`	`encap-limit`	Name change only
`rxrings`	`rx-rings`	Name change only
`txrings`	`tx-rings`	Name change only
`pfcmap-rmt`	`pfcmap-remote`	Name change only
`vsi-typeid`	`vsi-type-id`	Name change only
`vsi-vers`	`vsi-version`	Name change only
`vsi-mgrid`	`vsi-manager-id`	Name change only
`vsi-mgrid-enc`	`vsi-manager-id-encoding`	Name change only
`etsbw-lcl`	`ets-bw-local`	Name change only
`etsbw-rmt`	`ets-bw-remote`	Name change only
`vswitchmode`	`virtual-switching`	Name change only
`pvlan-tagmode`	`pvlan-tag-mode`	Name change only

New properties are supported by the `dladm` command that enable you to check speed negotiation of 40G-capable devices. You cannot set these properties with the `dladm set-linkprop` command. However, you can still display property information as usual.

For example, to check the 40G full-duplex setting for `ixgbe0`, type:

```
$ dladm show-linkprop -p adv-40gfdx-cap ixgbe0
```

To check whether an interface has 40G capability, display the `speed-duplex` property setting as follows:

```
$ dladm show-linkprop -p speed-duplex net8
LINK  PROPERTY PERM VALUE  EFFECTIVE  DEFAULT POSSIBLE
net8  speed-duplex   rw   40g-f   40g-f       --
```

See the `dladm`(8) man page.

SMF Management of Network Configuration

The following table describes the types of network configuration objects that have migrated to SMF. Configuration properties are stored within a property group that is named for that object, with each property group being of the type `application`.

TABLE 3 SMF Network Configuration Property Group Descriptions

Network Configuration Class or Object	Property Group Name	SMF Service
Datalinks	`datalinks`	`svc:/network/datalink-management:default`
Flows	`flows`	`svc:/network/datalink-management:default`
IP interfaces	`interfaces`	`svc:/network/ip-interface-management:default`
Protocol-related configuration (IP, UDP, TCP, SCTP, ICMP)	`protocols` Per-protocol properties are stored in `tcp`, `udp`, and `sctp`.	`svc:/network/ip-interface-management:default`
Static IP route configuration	`static-routes`	`svc:/network/ip-interface-management:default`
Known WLANs	`known-wlans` Property groups use the ESSID name (WLAN name) as the known WLAN property group name. Note that the security keys, which are stored in the `/etc/dladm/secobj.conf` file are not migrated to SMF.	`svc:/network/datalink-management:default`
ENMs	`enms` One property group represents each ENM	`svc:/network/physical:default`

Refer also to the `datalink-management(5)` and `ip-interface-management(5)` man pages.

Index

P

R

S

W