

# 在 Oracle® Solaris 11.2 中配置和管理网络组件

ORACLE®

文件号码 E53788  
2014 年 7 月



在 Oracle Solaris 11.2 中配置和管理网络组件

Part No: E53788

Copyright © 2011, 2014, Oracle and/or its affiliates. All rights reserved.

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

Référence: E53788

Copyright © 2011, 2014, Oracle et/ou ses affiliés. Tous droits réservés.

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 disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, 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.

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.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise 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 installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 des 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 ce type d'applications.

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 Xeon 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, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

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. 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.

# 目录

---

使用本文档 .....	9
1 关于 Oracle Solaris 中的网络管理 .....	11
Oracle Solaris 网络协议栈说明 .....	11
硬件层 .....	12
数据链路层 .....	12
网络层 .....	13
传输层 .....	13
应用程序层 .....	13
Oracle Solaris 网络协议栈中的命名和目录服务配置 .....	14
Oracle Solaris 中的网络设备和数据链路命名 .....	14
关于网络配置模式 .....	14
固定模式 .....	14
反应性模式 .....	15
关于基于配置文件的网络配置 .....	15
Oracle Solaris 网络管理命令 .....	15
dladm 命令 .....	16
ipadm 命令 .....	16
route 命令 .....	17
netcfg 和 netadm 命令 .....	17
用于重新配置系统网络的命令 .....	17
在网络上配置客户机系统所需的信息 .....	18
到何处查找有关 Oracle Solaris 中的网络管理的更多信息 .....	19
2 在 Oracle Solaris 中管理数据链路配置 .....	21
关于数据链路配置 .....	21
为数据链路指定通用名称 .....	22
定制操作系统指定通用链路名称的方式 .....	23
升级后的系统中的链路名称 .....	24
管理数据链路属性 .....	26

显示有关数据链路的常规信息 .....	26
显示系统的数据链路 .....	27
显示数据链路的物理属性 .....	27
删除数据链路 .....	28
重命名数据链路 .....	29
获取数据链路的运行时统计信息 .....	29
定制数据链路属性 .....	29
启用巨型帧支持 .....	30
修改链路速度参数 .....	31
设置数据链路上的 STREAMS 模块 .....	32
获取数据链路属性的状态信息 .....	32
其他 dladm 配置任务 .....	34
▼ 如何将 IP 配置从一个网络设备移动到另一个设备 .....	34
▼ 如何使用动态重新配置更换网络接口卡 .....	35
▼ SPARC: 如何确保每个接口的 MAC 地址是唯一的 .....	37
<b>3 在 Oracle Solaris 中配置和管理 IP 接口和地址 .....</b>	<b>39</b>
使用 ipadm 命令管理网络配置 .....	39
配置 IPv4 接口 .....	40
▼ 如何配置 IPv4 接口 .....	40
配置 IPv6 接口 .....	44
▼ 如何针对 IPv6 配置系统 .....	45
为 IPv6 接口使用临时地址 .....	46
配置 IPv6 标记 .....	49
在服务器上配置启用了 IPv6 的接口 .....	51
从 IPv4 网络迁移到 IPv6 网络 .....	51
配置路由 .....	52
路由表和路由类型 .....	53
创建持久性（静态）路由 .....	54
为单接口系统启用路由 .....	56
关于 IPv6 路由 .....	59
配置多宿主主机 .....	59
▼ 如何创建多宿主主机 .....	60
在多宿主主机上实现对称路由 .....	62
定制 IP 接口属性和地址 .....	63
设置 MTU 属性 .....	63
启用包转发 .....	63
定制 IP 地址属性 .....	64
禁用、删除和修改 IP 接口配置 .....	65

---

删除 IP 接口配置 .....	65
禁用 IP 接口配置 .....	66
删除或修改 IP 接口配置 .....	66
监视 IP 接口和地址 .....	67
获取有关 IP 接口的常规信息 .....	67
获取有关 IP 接口的信息 .....	68
获取有关 IP 接口属性的信息 .....	69
获取有关 IP 地址的信息 .....	70
获取有关 IP 地址属性的信息 .....	71
4 在 Oracle Solaris 客户机上管理命名和目录服务 .....	73
命名服务配置中的新增功能 .....	73
命名和目录服务配置概述 .....	74
关于 name-service/switch SMF 服务 .....	75
为本地文件模式配置系统 .....	76
▼ 如何以本地文件模式配置系统 .....	76
配置 DNS 客户机 .....	77
▼ 如何启用 DNS 客户机 .....	78
启用多播 DNS .....	79
为 DNS 通告资源 .....	79
配置 NIS 客户机 .....	80
▼ 如何在广播模式下配置 NIS 客户机 .....	80
▼ 如何使用特定的 NIS 服务器配置 NIS 客户机 .....	81
▼ 如何禁用 NIS 客户机服务 .....	81
配置 LDAP 客户机 .....	81
导入命名服务配置 .....	82
重置 SMF 命名服务配置 .....	82
5 关于在 Oracle Solaris 中管理基于配置文件的网络配置 .....	85
关于反应性模式 .....	85
关于基于配置文件的网络配置 .....	86
配置文件类型说明 .....	87
系统定义的配置文件和用户定义的配置文件 .....	90
使用基于配置文件的网络配置的准则 .....	90
配置文件激活策略 .....	91
配置文件激活模式 .....	91
使用基于配置文件的网络配置的安全要求 .....	92
基于配置文件的网络配置如何与其他 Oracle Solaris 功能一起使用 .....	93

6 在 Oracle Solaris 中管理基于配置文件的网络配置 .....	95
启用和禁用配置文件 .....	95
配置配置文件 .....	97
在 netcfg 交互模式下工作 .....	97
在 netcfg 命令行模式下使用 .....	98
在 netcfg 命令文件模式下使用 .....	99
创建 NCP .....	99
为 NCP 创建 NCU .....	100
创建位置 .....	103
创建 ENM .....	105
创建已知 WLAN .....	106
管理配置文件 .....	108
设置配置文件的属性值 .....	108
获取有关配置文件配置的信息 .....	110
使用 walkprop 子命令为配置文件设置属性值 .....	113
显示有关配置文件的属性值 .....	115
删除配置文件 .....	116
导出配置文件配置 .....	117
恢复导出的配置文件配置 .....	119
从桌面管理网络配置 .....	120
7 在 Oracle Solaris 中管理无线网络 .....	123
使用命令行管理无线网络 .....	123
▼ 如何连接到 WiFi 网络 .....	123
▼ 如何监视 WiFi 链路 .....	127
建立安全的 WiFi 通信 .....	128
▼ 如何通过指定 WEP 密钥来设置加密 WiFi 网络连接 .....	128
在反应性模式下管理已知 WLAN .....	130
从桌面管理无线网络 .....	130
▼ 如何连接无线网络 .....	131
从桌面管理收藏的无线网络 .....	131
索引 .....	133

## 使用本文档

---

- 概述 – 提供了有关在 Oracle Solaris 操作系统 (operating system, OS) 中如何配置和管理各种网络组件的信息，例如数据链路、IP 接口和地址、命名和目录服务、反应性配置文件和无线网络。
- 目标读者 – 负责管理公司数据中心内的网络配置的系统管理员。
- 必备知识 – 基本和高级网络管理概念和做法。

## 产品文档库

有关本产品的最新信息和已知问题均包含在文档库中，网址为：[http://www.oracle.com/pls/topic/lookup?ctx=product\\_intuitive\\_ID](http://www.oracle.com/pls/topic/lookup?ctx=product_intuitive_ID)。

## 获得 Oracle 支持

Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

## 反馈

可以在 <http://www.oracle.com/goto/docfeedback> 上提供有关此文档的反馈。



# ◆◆◆ 第 1 章

## 关于 Oracle Solaris 中的网络管理

---

本章概述了 Oracle Solaris 中组成主机客户机系统网络配置的各种组件。本书中介绍的任务和示例假定您是在安装之后执行网络配置。有关在安装期间配置网络的说明，请参见《[安装 Oracle Solaris 11.2 系统](#)》。

有关新增网络功能的信息，请参见《[Oracle Solaris 11.2 新增功能](#)》。

有关在网上配置客户机系统之前所需的规划任务，请参见《[在 Oracle Solaris 11.2 中规划网络部署](#)》。

有关常用网络管理命令的快捷方式，请参见《[Oracle Solaris 11.2 中的网络管理策略](#)》中的第 3 章, “[Oracle Solaris 网络管理命令速查表](#)”。

有关管理现有 TCP/IP 网络的信息，请参见《[在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道](#)》中的第 1 章, “[管理 TCP/IP 网络](#)”。

本章包含以下主题：

- “[Oracle Solaris 网络协议栈说明](#)” [11]
- “[Oracle Solaris 中的网络设备和数据链路命名](#)” [14]
- “[关于网络配置模式](#)” [14]
- “[关于基于配置文件的网络配置](#)” [15]
- “[Oracle Solaris 网络管理命令](#)” [15]
- “[在网上配置客户机系统所需的信息](#)” [18]
- “[到何处查找有关 Oracle Solaris 中的网络管理的更多信息](#)” [19]

## Oracle Solaris 网络协议栈说明

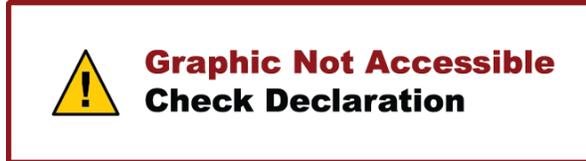
网络接口提供系统和网络之间的连接。这些接口是在数据链路上配置的，而这些数据链路又对应于系统中硬件设备的实例。

在 Oracle Solaris 10 中，存在对设备、数据链路和接口进行绑定的一对一关系，这意味着网络配置独立于硬件配置，也独立于网络拓扑。如果在硬件层实施了更改，例如更换网络接口卡 (network interface card, NIC)，则必须在系统上重新配置接口。

不过，在此 Oracle Solaris 发行版中，物理数据链路的命名不再绑定到与网络设备关联的底层硬件。缺省情况下，将为此类设备指定通用名称 net 和后缀，该后缀反映设备在

系统中的物理位置，如图 1-1 中所示。通过此分离机制，网络层上的网络配置不再绑定到硬件层中的芯片集或网络拓扑。

图 1 Oracle Solaris 11 网络协议栈



这种实现方式使网络管理更为灵活，表现在如下几个方面：

- 硬件层中可能出现的任何更改都不会影响网络配置。即使移除了底层硬件，链路和接口配置仍会保留。然后，这些相同的配置可以重新应用于任何更换的 NIC（只要这两个 NIC 的类型相同）。
- 通过实现网络配置与网络硬件配置的分离，可以在数据链路层中使用定制的链路名称。
- 使用抽象的数据链路层，多个抽象的网络概念或配置（例如虚拟局域网 (virtual local area network, VLAN)、虚拟网络接口卡 (virtual network interface card, VNIC)、物理设备、链路聚合和 IP 隧道）统一到了一个公共管理实体，即数据链路。

要比较 Oracle Solaris 10 网络栈与 Oracle Solaris 11 网络栈，请参见《[从 Oracle Solaris 10 转换至 Oracle Solaris 11.2](#)》中的“[将 Oracle Solaris 10 网络协议栈与 Oracle Solaris 11 网络协议栈进行比较](#)”。

## 硬件层

网络硬件设备也称为网络接口卡 (*network interface card, NIC*) 或网络适配器。NIC 可以是内置的，在购买系统时已存在于系统中。或者，您可以购买单独的 NIC 将其添加到系统。某些 NIC 卡只有单个接口。其他 NIC 品牌可能有多个接口，您可以配置这些接口来执行各种网络操作。

## 数据链路层

使用 `dladm` 命令在数据链路层执行网络配置。

下面是可在该层中执行的一些类型的数据链路配置：

- 物理链路的基本网络配置
- VNIC 的配置（基于物理链路的虚拟链路）  
对于 VNIC 配置，每个虚拟链路都具有自己的 MAC 地址

- 链路聚合配置  
聚合是基于物理链路配置的，用于提高稳定性和性能。
- 用于支持虚拟局域网 (virtual local area network, VLAN) 的 Etherstub
- 用于支持 VLAN 的网桥

有关其他示例，请参见《[在 Oracle Solaris 11.2 中管理网络数据链路](#)》。

## 网络层

使用 `ipadm` 命令在网络层执行网络配置。

下面是可在该层中执行的一些类型的 IP 配置：

- IP 接口配置，其名称与数据链路名称具有一对一的对应关系
- IP 网络多路径 (IP network multipathing, IPMP)
- 虚拟网络接口 (Virtual Network Interface, VNI)
- 对应于一个 IP 接口的多个 IP 地址
- 在单个 IP 接口上配置的 IPv4 和 IPv6 地址
- 使用其地址对象名称管理的 IP 地址

地址对象名称由接口名称后跟一个唯一的字符串组成，代表在系统上配置的 IP 地址。

对于其他示例，请参见下列参考信息：

- 《[在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道](#)》
- 《[在 Oracle Solaris 11.2 中管理网络虚拟化和网络资源](#)》
- 《[创建和使用 Oracle Solaris 区域](#)》

## 传输层

很少需要在传输层上执行显式配置。在 Oracle Solaris 上运行的应用程序通常会自动选择合适的传输协议以及对应的端口号。可以使用 `netstat` 命令查看活动端口。请参见 [netstat\(1M\)](#)。

可以使用 `ipadm` 命令优化某些传输协议参数。有关更多信息，请参见《[在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道](#)》中的“管理传输层服务”和《[Oracle Solaris 11.2 可调参数参考手册](#)》。

## 应用程序层

应用程序通过套接字 `xti` 或 `tli` 应用程序编程接口 (Application Programming Interface, API) 访问网络。在启动连接时，这些 API 需要客户机应用程序提供对应服务器的 IP 地

址和传输端口号。通常按照远程主机名称或服务名称而不是 IP 地址来识别服务器。应用程序使用标准库服务将主机名称和服务名称转换为 IP 地址，然后再尝试进行网络连接。

有关更多信息，请参见 [《Oracle Solaris 11.2 Programming Interfaces Guide》](#)。

## Oracle Solaris 网络协议栈中的命名和目录服务配置

客户机系统必须配置网络的名称和目录服务来执行 IP 地址查找。根据网络环境，配置过程可能在系统启动期间自动进行，也可能需要手动执行。此配置在网络栈的应用程序层进行。

有关客户机端命名和目录服务配置选项，请参见 [第 4 章 在 Oracle Solaris 客户机上管理命名和目录服务](#)。

## Oracle Solaris 中的网络设备和数据链路命名

一个数据链路代表开放系统互连 (Open Systems Interconnection, OSI) 模型的第二层 (L2) 中的一个链路对象。物理链路与设备直接关联，并拥有一个设备名称。设备名称包含驱动程序名称和设备实例编号。实例编号的值可以从 0 到  $n$ ，具体取决于系统上有多少 NIC 使用该驱动程序。

设备实例名称仍旧依赖于系统的底层硬件。但是，硬件层和软件层是分离的，这意味着在这些设备的基础上配置的数据链路不再和以前一样绑定。因此，为数据链路指定的名称可以不同于配置这些数据链路的底层设备名称。

缺省情况下，将为数据链路指定一个使用 net# 命名约定的通用名称，其中 # 是设备实例编号。此实例编号随系统中的每个设备递增，例如 net0、net1、net2 等。有关详细概述，请参见 [“关于数据链路配置” \[21\]](#)。

## 关于网络配置模式

Oracle Solaris 中支持两种网络配置模式：固定和反应性。

---

注 - 术语“固定”和“反应性”模式指系统是能够自动适应当前网络环境中的更改，还是在这些模式下使用时可以配置静态或固定 IP 地址。

---

### 固定模式

固定模式表示在系统上实例化的配置是持久的，不论网络条件如何变化都是如此。当发生这种更改（例如，添加接口）时，必须为系统重新配置网络以适应新环境。使用固定

模式时，每次使用相同的一组网络配置命令来配置您的系统。公司服务器由于网络环境相对稳定，通常会使用此配置模式。使用固定模式时，您使用 `dladm` 和 `ipadm` 命令管理网络配置的各个方面。请参见“[Oracle Solaris 网络管理命令](#)” [15]。

## 反应性模式

反应性模式则相反，网络将根据当前的网络条件自动进行配置。此模式主要用于手提电脑和笔记本个人电脑 (personal computer, PC) 以及网络条件可能会更改的情况。

在反应性模式中，网络守护进程 (nwmad) 监视系统的网络接口状态。只要网络条件发生更改，网络守护进程就会动态调整网络配置。例如，笔记本个人电脑可能会以物理方式连接到公司网络，也可能不以物理方式连接。以物理方式连接时，您很有可能会禁用笔记本电脑的无线接口。另外，在以太网电缆与笔记本电脑断开时，通常需要自动启用无线接口。此外，在切换到无线网络时，您可能希望系统自动调节 IP 过滤器设置。如果您采用反应性模式，则网络守护进程可以自动执行这些类型的动态配置更改。相反，在使用固定模式时，需要您手动完成重新配置步骤来执行这些类型的更改。

## 关于基于配置文件的网络配置

通过基于配置文件的网络配置可以定义多个备用配置，每个备用配置由一个配置文件（称为网络配置文件 (network configuration profile, NCP)）予以标识。例如，可以为笔记本个人电脑创建名为 `office` 的配置文件，该配置文件为系统配置静态 IP 地址和 DNS 服务器位置。备用 `home` 配置文件可以使用 DHCP 来获取此信息。通过单个命令，可在数秒内从一个配置文件切换到另一个配置文件。您可以启用的各种类型的配置文件支持两种可能的网络配置模式：固定和反应性。缺省模式是由系统上当前处于活动状态的配置文件决定的。

有关如何在 Oracle Solaris 安装期间在系统上激活配置文件的信息，请参见《[从 Oracle Solaris 10 转换至 Oracle Solaris 11.2](#)》中的“[安装期间如何配置网络](#)”。

如果不确定哪个配置文件当前在系统上处于活动状态，请使用 `netadm list` 命令来显示此信息。有关更多信息，请参见“[启用和禁用配置文件](#)” [95]。

有关 Oracle Solaris 中支持的各种类型的配置文件的完整说明，请参见第 5 章 [关于在 Oracle Solaris 中管理基于配置文件的网络配置](#)。

## Oracle Solaris 网络管理命令

下面是用于管理网络配置的命令：

- dladm
- ipadm
- route
- netcfg
- netadm

## dladm 命令

dladm 是在 Oracle Solaris 10 中引入的，用于配置数据链路。

dladm 命令用于管理以下类型的网络配置：

- 物理接口 - 以太网、无线和 InfiniBand
- 虚拟网络功能 - Etherstub、VNIC 和 IP 隧道
- 交换机功能 - 链路聚合、VLAN 和桥接技术
- 设备特性 - 速度、双工、优先级和功能协商

dladm 命令为系统上当前处于活动状态的配置文件创建持久性网络配置。因此，net0 在不同的配置文件中可以具有不同的 MTU 值。例如，如果为名为 net0 的数据链路配置了具体的最大传输单元 (Maximum Transmission Unit, MTU) 值 1200，则此 MTU 值仅在使用该配置文件时对 net0 具有持久性。如果随后激活了另一个配置文件并使用 dladm 命令为该配置文件设置了不同的 MTU 值，则新的 MTU 值将仅应用到该配置文件。请参见第 2 章 [在 Oracle Solaris 中管理数据链路配置](#)。

dladm 命令还取代了 Oracle Solaris 10 中用于配置协议属性的 ndd 命令。作为用于设置第 2 层驱动程序属性的工具，dladm 命令相比 ndd 命令具有多个优势。请参见《[从 Oracle Solaris 10 转换至 Oracle Solaris 11.2](#)》中的“[将 ndd 命令与 ipadm 命令进行比较](#)”。

## ipadm 命令

在此发行版中，ipadm 命令取代 ifconfig 命令用于配置 IP 接口和地址。ipadm 命令可更有效地管理 IP 接口和 IP 地址，因为该命令仅用于 IP 接口管理。此外，与 ifconfig 命令不同，ipadm 命令可实现持久性网络配置。请参见《[从 Oracle Solaris 10 转换至 Oracle Solaris 11.2](#)》中的“[将 ifconfig 命令与 ipadm 命令进行比较](#)”。

ipadm 命令还取代了 Oracle Solaris 10 中用于配置协议属性的 ndd 命令。作为用于设置协议属性的工具，ipadm 命令相比 ndd 命令具有多个优势。请参见《[从 Oracle Solaris 10 转换至 Oracle Solaris 11.2](#)》中的“[将 ndd 命令与 ipadm 命令进行比较](#)”。

## route 命令

由于 `/etc/defaultrouter` 文件在 Oracle Solaris 11 中已过时，因此您不能再使用此文件管理路由（缺省值或其他）。请改用 `route` 命令手动处理网络路由表。`route` 命令仅处理活动配置文件的路由。如果活动配置文件发生更改，则缺省路由以及其他路由可能会被替换。如果不在系统上切换配置文件，则不需要考虑这个问题。

有关更多信息，请参见[“创建持久性（静态）路由” \[54\]](#)。

## netcfg 和 netadm 命令

`netcfg` 和 `netadm` 命令用于管理各种类型的配置文件。这两个命令提供的大部分功能针对反应性配置文件的的管理。`netcfg` 命令很少在公司服务器上使用。这些类型的服务器通常使用固定模式。

`netadm` 命令用于启用和禁用配置文件以及显示有关配置文件及其状态的信息。请参见[“启用和禁用配置文件” \[95\]](#)和[“管理配置文件” \[108\]](#)。

---

注 - 通常使用 `netcfg` 命令配置反应性配置文件的属性。此外，如果配置文件当前处于活动状态，您还可以使用 `dladm` 和 `ipadm` 命令为反应性配置文件创建持久性配置。但是，不能使用 `netcfg` 命令配置系统唯一的固定配置文件 `DefaultFixed`。有关更多信息，请参见 [netcfg\(1M\)](#) 手册页。

---

## 用于重新配置系统网络的命令

可用于重新配置系统网络的另一个选项是 `sysconfig` 实用程序，也称为交互式系统配置 (System Configuration Interactive, SCI) 工具。SCI 工具支持配置全新安装或未配置的系统，它设计用来在文本安装期间为新创建的非全局区域提供系统配置。您能够以交互方式或非交互方式使用 SCI 工具。

使用 `sysconfig` 实用程序可以执行三种操作：取消配置、配置和配置文件创建。`unconfigure` 子命令用于取消配置整个系统。此命令会使系统保留在未配置状态。

`configure` 子命令用于重新配置整个系统或系统的一部分，这包括以下六个功能分组：

- network
- location
- users
- identity
- support
- kdb\_layout

有关已取消配置的分组的缺省值的更多信息，请参见《[安装 Oracle Solaris 11.2 系统](#)》中的“[功能分组概述](#)”。

例如，您将如下所示重新配置系统的现有命名服务：

```
# sysconfig configure -g network,naming_services
```

-g 选项用于指定应配置的特定功能分组。在本例中，配置了系统的网络组件。

有关更多信息，请参见 `sysconfig(1M)` 手册页和《[安装 Oracle Solaris 11.2 系统](#)》中的第 6 章,“[取消配置或重新配置 Oracle Solaris 实例](#)”。

## 在网络上配置客户机系统所需的信息

您可以在安装 Oracle Solaris 时或在安装 Oracle Solaris 之后配置网络。本书介绍了在安装之后在网络上配置客户机系统的任务。有关在安装期间配置网络的说明，请参见《[安装 Oracle Solaris 11.2 系统](#)》。

要在网络上配置客户机系统，必须提供以下信息：

- 主机名
- IP 地址
- 网络掩码

有关更多信息，请参见《[在 Oracle Solaris 11.2 中规划网络部署](#)》中的“[确定网络的 IP 地址寻址格式](#)”和《[在 Oracle Solaris 11.2 中规划网络部署](#)》中的“[获取网络的 IP 号](#)”。

如果网络被划分为多个子网，则您必须具有要应用于每个子网中的系统的子网号和 IP 地址架构，包括其各自的网络掩码。请参见《[在 Oracle Solaris 11.2 中规划网络部署](#)》中的“[在网络上使用子网](#)”。

- 系统所属的域名

请参见《[在 Oracle Solaris 11.2 中规划网络部署](#)》中的“[在网络上使用命名实体](#)”。

- 缺省路由器地址

如果在一个简单的网络拓扑中，每个网络仅连接有一个路由器，则可以提供此信息。请参见《[在 Oracle Solaris 11.2 中规划网络部署](#)》中的“[为网络规划路由器](#)”。

如果路由器不运行路由协议，如路由器搜索 (Router Discovery, RDISC) 服务器协议或路由器信息协议 (Router Information Protocol, RIP)，则也可以提供此信息。有关路由器的更多信息以及 Oracle Solaris 支持的路由协议列表，请参见《[将 Oracle Solaris 11.2 系统配置为路由器或负载均衡器](#)》中的“[路由协议](#)”。

在网络上配置客户机系统时，请参见《[在 Oracle Solaris 11.2 中规划网络部署](#)》中的“[IPv4 自治系统拓扑](#)”中的信息。

有关其中每个组件及相关任务的更多详细信息，请参见《[在 Oracle Solaris 11.2 中规划网络部署](#)》中的第 1 章,“[规划网络部署](#)”。

## 到何处查找有关 Oracle Solaris 中的网络管理的更多信息

除了本书中介绍的基本配置信息之外，请查阅以下参考资料，了解有关执行其他类型的网络管理的信息：

- 要定制网络协议和管理传输层服务，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的第 1 章,“管理 TCP/IP 网络”。
- 要将系统配置为路由器或者负载均衡器，请参见《将 Oracle Solaris 11.2 系统配置为路由器或负载均衡器》。
- 要执行高级数据链路和 IP 配置，包括链路聚合和各种类型的网桥，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》。
- 要配置 IPMP 组和 IP 隧道，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》。
- 要建立网络安全性，请参见《在 Oracle Solaris 11.2 中确保网络安全》。
- 要实现网络虚拟化功能，请参见《在 Oracle Solaris 11.2 中管理网络虚拟化和网络资源》。



## 在 Oracle Solaris 中管理数据链路配置

---

本章介绍了网络设备和数据链路，包括在使用固定模式时用于管理基本数据链路配置的任务。

以下信息仅与配置物理链路或者表示网络设备的链路相关。有关您可以配置的其他第 2 层 (Layer 2, L2) 实体的详细信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》和《在 Oracle Solaris 11.2 中管理网络虚拟化和网络资源》。

本章包含以下主题：

- “关于数据链路配置” [21]
- “管理数据链路属性” [26]
- “定制数据链路属性” [29]
- “其他 dladm 配置任务” [34]

### 关于数据链路配置

管理员在数据链路之上创建 IP 接口。每个数据链路代表开放系统互连 (Open Systems Interconnection, OSI) 模型的第二层中的一个链路对象。数据链路可以表示许多不同的 L2 实体，例如物理网络设备（称为物理链路）、物理数据链路的聚合、虚拟网络接口卡 (virtual network interface card, VNIC) 等。

链路名称是在自动创建关联的链路对象时指定的，也可以在创建数据链路时显式指定链路名称。物理链路（与物理网络设备关联的链路）是在添加设备时或者在安装之后首次引导 Oracle Solaris 系统时自动创建的。在此 Oracle Solaris 发行版中，物理数据链路的命名不再绑定到与网络设备关联的底层硬件。缺省情况下，为数据链路指定的名称以 net 为前缀，以反映数据链路在系统中物理位置的数字为后缀。例如，将为第一个板载网络设备 e1000g0 指定名称 net0，为下一个 e1000g1 设备指定名称 net1，依此类推。对于您显式创建的数据链路，例如链路聚合，您可以指定任意名称。此外，如果需要，您可以显式重命名为数据链路缺省指定的 netN 名称。

通用或灵活链路名称为网络配置提供了以下优点：

- 在单个系统内，动态重新配置 (dynamic reconfiguration, DR) 变得更容易。给定 NIC 的网络配置可以由不同的更换 NIC 继承。

- 区域迁移的网络设置不再过于复杂。如果目的地系统的链路与在迁移之前为区域指定的链路具有相同的名称，则在迁移后的系统中，该区域将保留其网络配置。因此，在迁移后无需对区域进行任何额外的网络配置。
- 通用命名约定使得在安装期间使用的系统配置 (System Configuration, SC) 清单中指定的网络配置不再过于复杂。由于所有系统的主网络数据链路通常都命名为 `net0`，对于为 `net0` 指定配置的多个系统，您可以使用一个通用 SC 清单。
- 数据链路管理也变得很灵活。您可以进一步定制数据链路的名称，例如，用来反映数据链路提供的特定功能。

下表说明了硬件 (NIC)、设备实例、链路名称以及链路上的接口之间的新对应关系。数据链路的名称由 OS 自动提供。

硬件 (NIC)	设备实例	链路的指定名称	IP 接口
e1000g	e1000g0	net0	net0
igb	ixgbe	net1	net1

如此表中所示，虽然设备实例名称仍基于硬件，但在安装之后，OS 对数据链路进行了重命名。

要显示数据链路、其通用名称以及对应设备实例之间的映射，请如下所示使用 `dladm show-phys` 命令：

```
# 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
```

## 为数据链路指定通用名称

- 物理网络设备是根据介质类型确定次序的，其中某些类型优先于其他类型。介质类型按优先级降序排序，如下所示：
  1. 以太网
  2. InfiniBand 设备
  3. 基于 IB 的以太网
  4. WiFi
- 在设备分组并按介质类型排序后，这些设备会根据其物理位置进一步排序，其中板载设备优先于外围设备。
- 对基于其介质类型和位置具有较高优先级的设备，会指定较低的实例编号。

在基于 SPARC 的系统上，将指定 netN 名称以匹配在 OpenBoot PROM (OBP) 中使用的 netN 设备别名。在基于 x86 的系统上，将使用 SMBIOS 数据（在可用时）来标识板载以太网设备并将其分配给 net0、net1 等。除了这些信息源之外（或者未在这些信息源中），位于较低的主板、IO 板、主桥、PCIe 根联合体、总线、设备和功能上的设备将排在其他设备前面，并且与位于较高的主板、主桥等等上的设备相比，将被分配较低的 net 实例。

要显示链路名称、设备和位置之间的对应关系，请如下所示使用 `dladm show-phys` 命令和 `-L` 选项：

```
# dladm show-phys -L
LINK          DEVICE          LOCATION
net0          e1000g0         MB
net1          e1000g1         MB
net2          e1000g2         MB
net3          e1000g3         MB
net4          ibp0            MB/RISER0/PCIE0/PORT1
net5          ibp1            MB/RISER0/PCIE0/PORT2
net6          eoib2           MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
net7          eoib4           MB/RISER0/PCIE0/PORT2/cloud-nm2gw-2/1A-ETH-2
```

## 定制操作系统指定通用链路名称的方式



注意 - 对自动指定通用链路名称的方式进行定制必须在安装 Oracle Solaris 之前进行。安装完成后，将无法在不删除现有配置的情况下定制缺省链路名称。

Oracle Solaris 使用前缀 net 来指定链路名称。不过，您可以使用您喜欢的任何定制前缀，例如 eth。您还可以禁用通用链路名称自动分配。

要禁用链路自动命名，或要定制链路名称前缀，请在系统配置 (System Configuration, SC) 清单中设置以下属性。SC 清单由 Oracle Solaris 的自动化安装程序 (Automated Installer, AI) 使用。

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

缺省情况下，`phys-prefix` 属性的值设置为 net，如上一个输出中的粗体文本所示。

- 要禁用自动命名，请将 `phys-prefix` 属性的值设置为空字符串，例如：

```
<propval name='phys-prefix' type='astring' value='' />
```

如果禁用了自动命名，数据链路名称将基于与其关联的硬件驱动程序，如 `bge0`、`e1000g0` 等。

- 要使用不同于 `net` 的前缀，请指定新前缀作为 `phys-prefix` 的值，如 `eth`。

如果为 `phys-prefix` 属性提供的值无效，则会忽略该值。数据链路将根据与其关联的硬件驱动程序来命名，如 `bge0`、`e1000g0` 等。有关有效链路名称的规则，请参见[“有效链路名称的规则” \[25\]](#)。

## 升级后的系统中的链路名称

在全新安装的系统上，数据链路将自动命名为 `net0` 到 `netN-1`，其中 `N` 表示网络设备的总数。

与此相反，如果是从其他 Oracle Solaris 11 发行版升级而来，则数据链路将保留在升级之前为其设置的名称。这些名称要么是缺省的基于硬件的名称，要么是在升级前由管理员为数据链路指定的定制名称。而且，在这些升级的系统上，随后添加的新网络设备也保留缺省的基于硬件的名称，而不是获得通用名称。升级后系统的这种行为确保不会有 OS 指定的通用名称与其他基于硬件的名称或管理员在升级前指定的定制名称混合使用。

您可以将基于硬件的名称和 OS 提供的链路名称替换为您喜欢使用的其他名称。通常情况下，由 OS 指定的缺省链路名称足够用于创建系统的网络配置。但是，在更改链路名称之前，请考虑以下信息。

### 替换基于硬件的链路名称

如果您系统的链路具有基于硬件的名称，至少要将这些链路重命名为通用名称。如果保留基于硬件的名称，以后在移除或更换这些物理设备时可能会出现混乱。

例如，您保留与设备 `bge0` 相关联的链路名称 `bge0`。并且在执行所有链路配置时都引用了该链路名称。之后，您可能将 NIC `bge` 更换为 NIC `e1000g`。要将以前设备的链路配置重新应用于新的 NIC `e1000g0`，您需要将链路名称 `bge0` 重新指定给 `e1000g0`。由于基于硬件的链路名称 `bge0` 与关联的 NIC `e1000g0` 不同，这种组合可能导致混乱。通过使用不基于硬件的名称，您可以更好地将链路与相关联的设备进行区分。

### 更改链路名称的注意事项

尽管替换基于硬件的链路名称是最佳做法，但在重命名链路之前，仍然必须进行周密的规划。更改设备的链路名称不会将新名称自动传播到所有现有的相关联配置。以下示例说明了更改链路名称时的风险：

- IP 过滤器配置中的一些规则应用于特定链路。更改链路名称时，这些过滤器规则会继续引用链路的原始名称。因此，在重命名链路之后，这些规则不再发挥预期的作用。您需要使用新链路名称调整过滤器规则，使其应用于链路。
- 请考虑导出网络配置信息的可能性。如前文所述，通过使用 OS 提供的 net# 缺省名称，您可以轻松地迁移区域并将网络配置导出到另一个系统。如果目标系统的网络设备是使用通用名称命名的，如 net0、net1 等，则区域将仅继承与指定给该区域的数据链路的名称相匹配的那些数据链路的网络配置。

因此，作为一般规则，不要随机地重命名数据链路。当重命名数据链路时，确保在更改链路名称后链路的所有关联配置继续适用。

一些可能会受链路重命名影响的配置如下：

- IP 过滤器规则
- 使用 ipadm 命令指定的 IP 配置
- Oracle Solaris 11 区域
- autopush 配置

---

注 - 当您重命名链路时，无需对 autopush 配置进行任何更改。但是，您必须了解在链路重命名后，配置使用每链路 autopush 属性的方式。有关更多信息，请参见《在 Oracle Solaris 11.2 中配置和管理网络组件》中的“设置数据链路上的 STREAMS 模块”。

---

## 有效链路名称的规则

指定链路名称时，请遵守以下规则：

- 链路名称必须包括一个字符串和一个物理连接点 (physical point of attachment, PPA) 编号。
- 链路名称必须遵守以下约束：
  - 名称最好包含 3 到 8 个字符。但是，名称最多可以有 16 个字符。
  - 名称的有效字符有字母数字 (a-z, 0-9) 和下划线 ('\_')。



注意 - 不要在链路名称中使用大写字母。

- 每个数据链路在某一时刻只能有一个链路名称。
- 每个数据链路在系统内必须具有唯一的链路名称。

---

注 - 作为附加限制，不能使用 lo0 作为灵活的链路名称。此名称保留用于标识 IP 回送接口。

---

指定链路名称时，链路在网络设置中的作用可能是很有用的参考信息。例如，可以将 `netmgt0` 指定给专用于网络管理的链路。将 `Upstream2` 用于连接到 ISP 的链路。作为一般规则，为了避免混淆，不要将已知设备的名称指定给您的链路。

## 管理数据链路属性

使用 `dladm` 命令定制通用数据链路属性可以提供以下优点：

- `dladm` 命令是在配置网络驱动程序属性时唯一需要的命令接口。此命令取代了以前通过使用 `ndd` 命令并修改 `driver.conf` 文件来设置驱动程序属性的做法。
- 不论设置什么属性，都请使用以下统一语法：  
`dladm subcommand properties datalink`
- `dladm` 命令对驱动程序的公共属性和专用属性均适用。
- 对特定驱动程序使用 `dladm` 命令不会中断相似类型的其他 NIC 的网络连接。因此，您可以动态地配置数据链路属性。
- 数据链路配置值存储在 `dladm` 系统信息库中，在系统重新引导后仍会保留。

## 显示有关数据链路的常规信息

不带任何选项使用时，`dladm` 命令将显示有关系统的数据链路的常规信息，包括种类、状态和底层物理链路。

```
# 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
```

数据链路可以是物理链路以外的其他种类，例如链路聚合、虚拟 LAN (virtual LAN, VLAN) 和虚拟 NIC (virtual NIC, vNIC)。这些其他数据链路也包含在 `dladm` 命令显示的缺省信息中。例如，在以上输出中，物理链路 `net1` 和 `net4` 上配置了链路聚合 (`aggr0`)。

有关链路聚合和 VLAN 的信息，请参见 [《在 Oracle Solaris 11.2 中管理网络数据链路》](#)。有关 vNIC 的信息，请参见 [《在 Oracle Solaris 11.2 中管理网络虚拟化和网络资源》](#)。

## 显示系统的数据链路

可以使用 `dladm show-link` 命令同时显示系统上的物理数据链路和虚拟数据链路。系统包含的数据链路与已安装的 NIC 一样多。您可以将各个选项与此命令一起使用来定制所显示的信息。

不带其他选项或参数使用时，`dladm show-link` 命令显示以下信息：

```
# 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      --
```

在以上输出中，STATE 列显示了虚拟数据链路的当前状态。此状态可能为 up、down 或 unknown。对于虚拟数据链路，当一个 NIC 拆分为多个 VNIC 时，将在内部隐式创建虚拟交换机。创建的此虚拟交换机使得 VNIC 和主数据链路只要位于相同的 VLAN 上就能够彼此通信，即使物理数据链路没有连接到外部网络也是如此。这种关系形成了数据链路的虚拟状态。

使用 `-P` 选项可以显示有关数据链路的持久性配置信息。基于该命令提供的信息，您可以进一步配置网络。例如，您可以确定系统上 NIC 的数量，然后可以选择要使用的数据链路，在该数据链路上，您可以配置 IP 接口。键入命令时，显示的信息类似于以下示例：

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

以上示例显示了系统具有三个与对应的物理 NIC 直接相关联的数据链路。不存在任何特殊数据链路，如在 `phys` 类下的数据链路上配置的聚合或虚拟 NIC。

## 显示数据链路的物理属性

使用 `dladm show-phys` 命令可获取与物理 NIC（与系统数据链路相关联）相关的系统数据链路的信息。不带任何选项使用时，该命令将显示类似于以下示例的信息：

```
# 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
```

除了其他详细信息，以上输出还显示了与具有通用链路名称的数据链路相关联的物理 NIC。例如，net0 是 NIC e1000g0 的数据链路名称。要显示有关已为数据链路设置的标志的信息，请使用 -P 选项。例如，标记有 r 的数据链路意味着其底层 NIC 已被移除。

在以上输出中，STATE 列显示了物理数据链路的当前状态。此状态可能为 up、down 或 unknown。物理链路状态表示物理设备是否连接到外部网络（如果插入了电缆，并且电缆另一端的端口的状态为 up，则表明已连接到外部网络）。

-L 选项是您可以使用的另一个很有用的选项。此选项显示每个数据链路的物理位置。位置确定数据链路的实例编号，例如 net0、net1 等。

```
# 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
```

使用 -m 选项可显示系统中物理链路的 MAC 地址：

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

此命令类似于使用 ifconfig 命令。

显示系统中所有链路（包括物理的和非物理的）的 MAC 地址，如下所示：

```
# 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
--
```

## 删除数据链路

可以使用 `dladm delete-phys` 命令从系统中删除数据链路。

删除数据链路与移除物理 NIC 没有紧密联系。例如，如果从系统中移除了物理 NIC，则与该 NIC 关联的数据链路配置仍保留，因为软件层不再绑定到硬件层，如《[从 Oracle Solaris 10 转换至 Oracle Solaris 11.2](#)》中的“[将 Oracle Solaris 10 网络协议栈与 Oracle Solaris 11 网络协议栈进行比较](#)”中所述。因此，您仍然可以通过将该数据链路的名称指定给其他 NIC 的关联链路来在其他底层物理 NIC 中使用该数据链路配置。

如果您要分离 NIC 而不更换它，且不再需要其数据链路配置，则可以如下所示删除该数据链路：

```
# dladm delete-phys datalink
```

---

提示 - 要确认数据链路的 NIC 是否已移除，请使用 `dladm show-phys -P` 命令。此输出提供了一个 `FLAGS` 列，在此列中，`r` 标志指明与物理链路相关联的物理设备是否已移除。

---

## 重命名数据链路

可以使用 `dladm rename-link` 命令重命名数据链路。在 Oracle Solaris 系统上，OS 会自动向所有数据链路提供通用名称。有关通用数据链路名称的更多信息，请参见[“关于数据链路配置” \[21\]](#)。

缺省情况下，这些通用名称使用命名格式 `net $n$` ，例如 `net0`、`net1`、`net2` 等。由于 OS 管理着这些名称，因此您无法通过执行常规的管理任务来重命名数据链路。有关更改链路名称所需的过程，请参见[如何将 IP 配置从一个网络设备移动到另一个设备 \[34\]](#)。

## 获取数据链路的运行时统计信息

可以使用 `dlstat` 命令获取所有类型的数据链路的运行时数据链路统计信息。仅使用命令本身不使用其他选项时，`dlstat` 显示系统上所有数据链路的相关统计信息，如以下输出中所示：

```
% dladm
LINK      IPKTS    RBYTES   OPKTS    OBYTES
net0      58.00K   9.52M    5.61K    1.91M
```

有关使用 `dlstat` 命令的更多信息，请参见《[在 Oracle Solaris 11.2 中管理网络虚拟化和网络资源](#)》中的第 8 章，“[监视网络通信流量和资源使用情况](#)”。另请参见 `dlstat(1M)` 手册页。

## 定制数据链路属性

除了执行基本的数据链路配置，您还可以使用 `dladm` 命令设置数据链路属性并根据您的网络要求对其进行定制。

下面三个 `dladm` 子命令用于管理数据链路属性：

<code>dladm show-linkprop -p <i>propertydata</i>link</code>	显示数据链路的属性及其当前值。如果不使用 <code>-p <i>property</i></code> 选项，则将显示数据链路的所有属性。如果不指定数据链路，则将显示所有数据链路的所有属性。
---	--

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

为数据链路的属性指定值。

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

将数据链路的特定属性重置为其缺省值。

您可以定制的数据链路属性取决于特定 NIC 驱动程序支持的属性。

可通过使用 `dladm` 命令配置的数据链路属性分为两个类别：

- 公共属性 – 这些属性可应用于给定介质类型的任何驱动程序，例如链路速度、以太网的自动协商，或可应用于所有数据链路驱动程序的最大传输单元 (maximum transmission unit, MTU) 大小。
- 专用属性 – 这些属性特定于给定介质类型的 NIC 驱动程序的特定子集。这些属性可能是特定于该子集的，因为它们与驱动程序相关联的硬件密切相关，或者与驱动程序实现本身的详细信息（如与调试相关的可调参数）密切相关。

链路属性通常具有缺省值。但是，在某些联网情况下，可能需要您更改特定属性值。例如，NIC 可能会与不能正确执行自动协商的旧交换机进行通信。或者，一个交换机可能已配置为支持巨型帧。或者，可能需要为特定驱动程序修改用于控制数据包传输或数据包接收的特定于驱动程序的属性。

## 启用巨型帧支持

MTU 定义协议可从系统中传输的最大包的大小。缺省情况下，大部分 NIC 驱动程序将 MTU 大小定义为 1500。但是，如果有巨型帧穿越网络，则缺省值将不够用。要支持巨型帧，MTU 大小至少需要为 9000。

---

注 - 数据链路和 IP 接口都有 MTU 属性，这意味着您可以为数据链路使用一个 MTU 值，为在该链路上配置的 IP 接口使用另一个 MTU 值。数据链路 MTU 的值影响您可以为 IP 接口的 MTU 设置的可能值。有关为数据链路和 IP 接口配置 MTU 属性时此行为的影响的更多信息，请参见“[设置 MTU 属性](#)” [63]。

---

如下所示更改 MTU 大小的缺省值：

```
# dladm set-linkprop -p mtu=new-size datalink
```

更改 MTU 大小后，您可以在数据链路上重新配置一个 IP 接口。

以下示例显示了如何启用对巨型帧的支持。此示例假定您已删除了数据链路上现有的任何 IP 接口配置。

```
# 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         --          --
```

## 修改链路速度参数

大多数网络设置包含多个具有不同速度能力的系统。每个系统均会将速度能力通告给网络中的其他系统，网络将通知每个系统如何传输和接收网络通信。

以下成对的数据链路属性控制系统通告的速度能力：

- `adv_10gfdx_cap/en_10gfdx_cap`
- `adv_1000fdx_cap/en_1000fdx_cap`
- `adv_1000hdx_cap/en_1000hdx_cap`
- `adv_100fdx_cap/en_100fdx_cap`
- `adv_100hdx_cap/en_100hdx_cap`
- `adv_10fdx_cap/en_10fdx_cap`
- `adv_10hdx_cap/en_10hdx_cap`

每个链路速度能力均涉及一对属性：通告速度 (`adv_*_cap`) 和支持的通告速度 (`en_*_cap`)。此外，还将为全双工和半双工能力提供数据链路速度信息，如属性名称中的 `*fdx*` 和 `*hdx*` 所指定。通告速度属性为只读属性，指示特定数据链路速度是否已通告。您可以通过设置相应的 `en_*_cap` 属性来确定是否通告了某个特定的数据链路速度。

缺省情况下，将通告数据链路的所有速度和双工能力。但是，可能会存在以下情况：新系统正在与旧系统通信，但是已禁用了或不支持自动协商。要启用这两个系统间的通信，较旧系统与较新系统之间的通告速度可能需要更改为较低的值。系统的千兆传输能力可能需要关闭，并且只通告较低的速度能力。在这种情况下，您应键入以下命令来为全双工能力和半双工能力关闭千兆传输能力的通告。

```
# dladm set-linkprop -p en_1000fdx_cap=0 datalink
# dladm set-linkprop -p en_1000hdx_cap=0 datalink
```

要显示这些属性的新值，请如下所示使用 `dladm show-linkprop` 命令：

```
# dladm show-linkprop -p adv_10gfdx_cap datalink
# dladm show-linkprop -p adv_1000hdx_cap datalink
```

通常情况下，给定的支持的速度属性和相应通告属性的值是相同的。不过，如果 NIC 支持一些高级功能（如电源管理），这些功能可能对主机和其链路伙伴之间实际通告的位设置限制。例如，使用电源管理时，`adv_*_cap` 属性的设置可能只是 `en_*_cap` 属性的设置的子集。

## 设置数据链路上的 STREAMS 模块

当数据链路处于打开状态时，可以设置将最多 8 个 STREAMS 模块推送到流。这些模块通常由第三方网络软件（如虚拟专用网络 (virtual private networks, VPN) 和防火墙）使用。软件供应商会提供有关此类网络软件的文档。

要在特定数据链路上推送的模块的列表是由 `autopush` 属性控制的。而 `autopush` 属性的值是通过使用 `dladm set-linkprop` 命令设置的。

可以使用单独的 `autopush` 命令基于每个驱动程序将模块推送到数据链路的流。此命令使用为每个驱动程序设置的配置文件，配置文件会告诉命令要推送的模块。然而，驱动程序始终绑定到 NIC。如果移除了数据链路的底层 NIC，则链路的 `autopush` 属性信息也会丢失。

因此，将 `dladm` 命令用于此目的比使用 `autopush` 命令更加可取。对于某个特定的数据链路，如果同时存在每驱动程序和每链路 `autopush` 配置，则将使用通过 `dladm set-linkprop` 命令设置的每链路信息，而忽略每驱动程序信息。

要在数据链路处于打开状态时将模块推送到 STREAMS，您可以使用相同的 `dladm set-linkprop` 命令为 `autopush` 属性指定模块。例如，可以如下所示在 `net0` 链路上推送 `vpnmod` 和 `bufmod` 模块：

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

## 获取数据链路属性的状态信息

要获取有关数据链路属性的信息，可以使用以下任一命令：

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

## 显示数据链路属性

要显示数据链路属性的完整列表，请键入该命令而不指定属性，如以下示例中所示：

```
# dladm show-linkprop net1
LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
net1      speed         r-  0           0             0           --
net1      autopush      rw  --          --           --          --
net1      zone         rw  --          --           --          --
net1      duplex       r-  unknown     unknown       unknown     half,full
net1      state        r-  up          up            up          up,down
net1      adv_autoneg_cap --  --          --            0           1,0
```

```

net1      mtu          rw    1500      1500      1500      1500
net1      flowctrl       -- --          --          no         no,tx,rx,bi,
                                pfc,auto
net1      adv_10gfdx_cap r- --          --          0          1,0
net1      en_10gfdx_cap -- --          --          0          1,0
net1      adv_1000fdx_cap r- --          --          0          1,0
net1      en_1000fdx_cap -- --          --          0          1,0
net1      adv_1000hdx_cap r- --          --          0          1,0
net1      en_1000hdx_cap -- --          --          0          1,0
net1      adv_100fdx_cap r- --          --          0          1,0
net1      en_100fdx_cap -- --          --          0          1,0
net1      adv_100hdx_cap r- --          --          0          1,0
net1      en_100hdx_cap -- --          --          0          1,0
net1      adv_10fdx_cap r- --          --          0          1,0
net1      en_10fdx_cap -- --          --          0          1,0
net1      adv_10hdx_cap r- --          --          0          1,0
net1      en_10hdx_cap -- --          --          0          1,0
net1      maxbw       rw --          --          --         --
net1      cpus        rw --          --          --         --

```

## 显示以太网属性值

如果没有随 `dladm show-ether` 命令指定任何选项，则只会显示数据链路的当前以太网属性值。要获取缺省提供的内容之外的信息，请使用 `-x` 选项，如以下示例所示：

```

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

```

如果使用 `-x` 选项，该命令还显示指定链路的内置能力，以及主机和链路伙伴之间当前通告的能力。

本例中显示了以下信息：

- 对于以太网设备的当前状态，该链路处于活动状态并在全双工模式下以每秒 1 Gb 速度工作。其自动协商功能已启用并采用双向流量控制，即主机和链路伙伴都可以发送和接收 PAUSE 帧。此信息显示在输出的第一行中。
- 示例输出中后面的行显示有关数据链路速度能力、已通告的实际数据链路速度以及来自对等方系统的信息，如下所示：
  - 列出以太网设备的功能。协商类型可以设置为自动。此外，该设备可以支持的速度有每秒 1 Gb、每秒 100 Mb 和每秒 10 Mb（全双工和半双工）。同样，主机和链路伙伴之间可以双向接收或发送 PAUSE 帧。
  - `net1` 的功能按如下所示通告：自动协商、速度双工和 PAUSE 帧的流控制。
  - 同样，`net1` 的链路或对等伙伴通告以下功能：自动协商、速度双工和 PAUSE 帧的流控制。

## 其他 dladm 配置任务

本节介绍了因使用 dladm 命令而简化的其他配置过程，如切换主接口和执行动态重新配置 (dynamic reconfiguration, DR)。

### ▼ 如何将 IP 配置从一个网络设备移动到另一个设备

如果需要保留与一个网络设备关联的 IP 配置，然后将该配置移动到另一个网络设备，请使用以下过程。可以执行此过程作为从系统中移除网卡或者更改网络电缆连接的前奏。

此过程仅用于示范目的，它介绍了如何保留与 net0 (e1000g0) 设备关联的 IP 配置，然后将配置应用于 nge0 设备。

1. 成为管理员。
2. 显示系统上的物理链路到设备映射。

以下示例假定 e1000g0 的 IP 配置由于某种原因已关闭，因此需要将配置移动到 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
```

3. 临时禁用数据链路上的 IP 配置，保留其持久性设置不变。

```
# ipadm disable-if interface
```

例如，您可以如下所示禁用 net0 上的 IP 配置：

```
# ipadm disable-if net0
```

通过此步骤，您可以重命名数据链路而无需重新创建其 IP 配置。

4. 重命名数据链路。
- 例如，可以如下所示重命名 net0 数据链路：

```
# dladm rename-link net0 oldnet0
```

5. 将主链路名称指定给要指派为主设备的数据链路。

```
# dladm rename-link new-link primary-link
```

例如，您可以如下所示将 net0 链路名称重新指定给 net2 数据链路：

```
# dladm rename-link net2 net0
```

6. 重新启用新命名的数据链路路上的 IP 配置。例如：

```
# ipadm enable-if -t net0
```

#### 例 1 删除数据链路接口

执行全新安装时，将使用命名约定 net0、net1 和 netN 等（取决于系统上的网络设备总数）自动为系统上的所有数据链路指定通用名称。安装后，可以为数据链路指定其他名称。以下示例显示了如何更改最初为接口提供的 IP 地址，这涉及到首先删除现有接口：

```
# ipadm delete-ip net0
# ipadm create-ip net0
# ipadm create-addr -T addrconf net0/new-add
```

有关更多信息，请参见《在 Oracle Solaris 11.2 中配置和管理网络组件》中的第 2 章，“在 Oracle Solaris 中管理数据链路配置”。

## ▼ 如何使用动态重新配置更换网络接口卡

以下过程仅适用于支持动态重新配置 (dynamic reconfiguration, DR) 的系统。该过程专门介绍了完成 DR 后的配置步骤。完成 DR 过程之后，您不再需要重新配置网络链路。相反，您只需将已删除 NIC 的链路配置传输至更换 NIC。

该过程未介绍执行 DR 本身的步骤。有关该信息，请查阅您的系统文档。

有关 DR 的介绍，请参见《在 Oracle Solaris 11.2 中管理设备》中的第 2 章，“动态配置设备”。

开始之前 务必先完成以下步骤：

- 确保您的系统支持 DR。
- 参考描述您系统上的 DR 的相应手册。

要查找有关 Oracle 提供的 Sun 服务器上的 DR 的最新文档，请在 <http://www.oracle.com/technetwork/indexes/documentation/index.html> 中搜索“动态重新配置”。

有关在 Oracle Solaris Cluster 环境中执行 DR 的信息，请参见《Oracle Solaris Cluster 系统管理指南》。

1. 成为管理员。
2. (可选) 显示有关数据链路的物理属性及其在系统上的各自位置的信息。

```
# dladm show-phys -L
```

有关 `dladm show-phys -L` 命令显示的信息类型的更多信息，请参见 [dladm\(1M\)](#) 手册页。

3. 按照您的系统文档中的说明执行 DR 过程。
4. 在安装了替换的 NIC 之后，根据所适用的环境，按照如下所述继续操作：
  - 如果您将替换的 NIC 插入旧 NIC 原来所在的插槽，请继续执行步骤 5。  
如果新 NIC 使用旧 NIC 先前占用的位置，则新 NIC 会继承旧 NIC 的链路名称和配置。
  - 如果您将替换的 NIC 插入到其他插槽中，且新 NIC 需要继承已删除 NIC 的数据链路配置，请按如下所示重命名链路：

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

*new-datalink* 指的是位于其他插槽（不同于移除旧 NIC 的位置）中替换的 NIC 的数据链路。

*old-datalink* 指的是与移除的旧 NIC 相关联的数据链路名称。

---

注 - 在此方案中，已从中移除旧 NIC 的插槽必须留空。

---

例如，插槽 1 中的 NIC 已移除，然后在插槽 2 中插入了新 NIC。插槽 1 中未插入 NIC。假定插槽 1 中的数据链路是 `net0`，插槽 2 中的数据链路是 `net1`。您可以如下所示指定新 NIC 的数据链路继承旧 NIC 数据链路配置：

```
# dladm rename-link net1 net0
```

5. 通过启用新 NIC 的资源以便其可供使用，完成 DR 过程。  
例如，您可以使用 `cfgadm` 命令配置 NIC。有关更多信息，请参见 [cfgadm\(1M\)](#) 手册页。
6. (可选) 显示链路信息。  
可以使用 `dladm show-phys` 命令或 `dladm show-link` 命令显示有关数据链路的信息。

#### 例 2 通过安装新的网卡执行动态重新配置

以下示例显示了如何使用 `e1000g` 卡更换具有链路名称 `net0` 的 `bge` 卡。在 `e1000g` 连接到系统后，`net0` 的链路配置将从 `bge` 传输到 `e1000g`。

```
# 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
```

您将执行特定于 DR 的步骤，例如使用 `cfgadm` 命令删除 bge 卡，然后在其位置安装 e1000g 卡。安装该卡后，e1000g0 的数据链路会自动采用名称 net0 并继承链路的配置。

```
# 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: 如何确保每个接口的 MAC 地址是唯一的

每个基于 SPARC 的系统都有一个系统级 MAC 地址，缺省情况下所有接口都使用该地址。但是，某些应用程序要求主机上的每个接口都有唯一的 MAC 地址。类似地，某些类型的接口配置（如链路聚合和 IP 网络多路径 (IP multipathing, IPMP)）也要求接口必须具有自己的 MAC 地址。

EEPROM 参数 `local-mac-address?` 决定了基于 SPARC 的系统上的所有接口是使用系统级 MAC 地址还是使用唯一 MAC 地址。以下过程介绍了如何使用 `eeprom` 命令检查 `local-mac-address?` 参数的当前值以及如何根据需要更改该参数的值。

1. 成为管理员。
2. 确定系统上的所有接口当前是否都使用系统级 MAC 地址。

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

在以上输出中，`local-mac-address?=false` 设置指示所有接口都使用系统级 MAC 地址。例如，必须先将 `local-mac-address?=false` 设置的值更改为 `local-mac-address?=true`，然后任何接口才能成为 IPMP 组的成员。

---

注 - 配置链路聚合时，也应进行此更改。

---

3. 如下所示更改 `local-mac-address?` 设置的值：

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

重新引导系统时，具有出厂时安装的 MAC 地址的接口将使用这些出厂设置，而不使用系统级 MAC 地址。没有出厂时安装的 MAC 地址的接口将继续使用系统级 MAC 地址。

4. 检查系统上所有接口的 MAC 地址。

查找多个接口具有相同 MAC 地址的情况。在以下示例中，有两个接口使用系统级 MAC 地址 8:0:20:0:0:1。

```
# 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 --
```

5. (可选) 如有必要，请手动配置其余的接口，以便所有接口都具有唯一的 MAC 地址。

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

---

注 - 仅当有两个或两个以上的网络接口具有相同 MAC 地址时，才需要执行此步骤。

---

在以上示例中，您需要为 net0 和 net1 配置在本地管理的 MAC 地址。例如，要用本地管理的 MAC 地址 06:05:04:03:02 重新配置 net0，您应输入以命令：

```
# dladm set-linkprop -p mac-address=06:05:04:03:02 net0
```

6. 重新引导系统。

## 在 Oracle Solaris 中配置和管理 IP 接口和地址

---

本章介绍如何配置实施了 IPv4 和 IPv6 寻址的网络。本章中的许多任务同时适用于启用了 IPv4 和启用了 IPv6 的网络。仅与 IPv4 或 IPv6 网络相关的过程将会具体指明。

在配置网络之前，请复查《在 Oracle Solaris 11.2 中规划网络部署》中所述的与 IP 相关的规划任务。

有关管理其他 TCP/IP 属性（例如全局包转发和传输层服务）的相关信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的“管理传输层服务”。

本章包含以下主题：

- “使用 ipadm 命令管理网络配置” [39]
- “配置 IPv4 接口” [40]
- “配置 IPv6 接口” [44]
- “从 IPv4 网络迁移到 IPv6 网络” [51]
- “配置路由” [52]
- “配置多宿主主机” [59]
- “定制 IP 接口属性和地址” [63]
- “定制 IP 地址属性” [64]
- “禁用、删除和修改 IP 接口配置” [65]
- “监视 IP 接口和地址” [67]

### 使用 ipadm 命令管理网络配置

引入了 ipadm 命令来最终取代 ifconfig 命令作为配置 IP 接口的主要方式。

ipadm 命令还取代了 ndd 命令用于配置以下 TCP/IP 协议的属性：

- IP
- 地址解析协议 (Address Resolution Protocol, ARP)
- 流控制传输协议 (Stream Control Transmission Protocol, SCTP)
- Internet 控制消息协议 (Internet Control Message Protocol, ICMP)
- 上层协议，例如 TCP 和用户数据报协议 (User Datagram Protocol, UDP)

作为用于配置接口的工具，`ipadm` 命令相比 `ifconfig` 命令具备以下优势：

- 提供了一个面向对象的子命令结构，该结构优于 `ifconfig` 命令提供的结构。这一更改最终使得网络配置过程更容易理解。
- 与 `ifconfig` 命令不同的是，它能够使网络配置更改成为持久性的。
- 支持一个可解析的输出选项，这在脚本编写时非常有用。

作为设置协议属性的工具，`ipadm` 命令与 `ndd` 命令相比，具有以下优点：

- 相比 `ndd` 命令提供了更广泛的属性信息，例如，属性的当前值和缺省值，以及可能值的范围。
- 持久还是临时设置属性值。`ndd` 命令仅临时设置属性值。
- 支持一个可解析的输出选项，这在脚本编写时非常有用。

要将 `ipadm` 命令与 `ifconfig` 和 `ndd` 命令进行比较，请参见 [《从 Oracle Solaris 10 转换至 Oracle Solaris 11.2》](#) 中的“网络管理命令更改”。

有关 `ipadm` 命令的更多信息，请参见 [ipadm\(1M\)](#) 手册页。

## 配置 IPv4 接口

以下过程和示例介绍了如何配置使用 IPv4 地址的网络。

### ▼ 如何配置 IPv4 接口

**开始之前** 检查在系统上处于活动状态的 NCP 以确保您将配置应用于正确的配置文件。请参见 [例 18 “在固定模式和反应性模式之间切换”](#)。

1. 成为管理员。
2. 创建接口。

```
# ipadm create-interface-class interface
```

`interface-class` 指您可以创建的以下三类接口之一：

- IP 接口  
此接口类是您在执行网络配置时创建的最常见的类。要创建此接口类，请使用 `create-ip` 子命令。
- STREAMS 虚拟网络接口 (virtual network interface, VNI 接口)

要创建此接口类，请使用 `create-vni` 子命令。

从 Oracle Solaris 11.2 开始，您可以更随意地命名 VNI 接口。以前，VNI 接口名称中必须包括前缀 "vni"，例如 `vni0`。现在不再有此要求。有关 VNI 设备和接口的更多信息，请参见 [vni\(7d\)](#) 和 [ipadm\(1M\)](#) 手册页。

- IPMP 接口

在配置 IPMP 组时使用此接口类。要创建此接口类，请使用 `create-ipmp` 子命令。有关 IPMP 组的更多信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的第 2 章,“关于 IPMP 管理”。

*interface*                    指接口的名称。该名称与用于创建接口的数据链路的名称相同。要显示系统上的数据链路，请使用 `dladm show-link` 命令。

### 3. 通过使用以下命令之一配置具有有效 IP 地址的 IP 接口：

- 配置静态 IP 地址：

```
# ipadm create-addr -a address [interface | addrobj]
```

`-a address`                    指定要在接口上配置的 IP 地址。

---

注 - 隧道配置通常需要隧道接口的两个地址：一个本地地址和一个远程地址。有关本地和远程地址以及隧道配置的信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的第 5 章,“管理 IP 隧道”。

---

对于数字 IP 地址，请使用无类域间路由 (Classless Inter-Domain Routing, CIDR) 表示法。如果不使用 CIDR 表示法，则将使用 `svc:/system/name-service/switch:default` 网络掩码数据库搜索顺序或使用有类地址语义来确定网络掩码。

您也可以选择指定主机名，而非数字 IP 地址。如果在 `/etc/hosts` 文件中为主机名定义了相应的数字 IP 地址，则使用该主机名是有效的。如果未在文件中定义数字 IP 地址，则该数值可通过使用在 `name-service/switch` 服务中为 `host` 指定的解析器顺序唯一获取。如果给定的主机名存在多项，则会生成错误。

---

注 - 在引导过程中，将先配置 IP 地址，再使命名服务联机。因此，您必须确保在 `/etc/hosts` 文件中定义在网络配置中使用的任何主机名。

---

[*interface* | *addrobj*] 在 Oracle Solaris 中，每个地址由一个相应的地址对象予以标识，在命令中由 *addrobj* 表示。对于此地址的任何后续配置，您可以参考地址对象而非实际 IP 地址。例如，您可以键入 `ipadm show-addr addrobj` 或 `ipadm delete-addr addrobj`。要自动生成地址对象名称，请仅指定 *interface* 的接口名称。要手动命名地址对象，请直接提供地址对象名称。

- 如果您指定接口名称，则地址对象将以 *interface/address-family* 格式自动命名。*Address family* 是 IPv4 地址的 v4 或 IPv6 地址的 v6。如果使用自动生成的地址对象名称在接口上配置了多个地址，则会在地址对象名称中附加字母以使其唯一。例如，`net0/v4`、`net0/v4a`、`net0/v4b`、`net0/v6`、`net0/v6a` 等。
  - 如果您手动命名 *addrobj* 的地址对象，则必须使用格式 *interface/user-specified-string*。*User-specified-string* 表示由字母数字字符组成的字符串，该字符串以字母开头且最大长度为 32 个字符。例如，您可以命名地址对象 `net0/static`、`net0/static1`、`net1/private` 等。
- 配置非静态地址。

```
# ipadm create-addr -T address-type [interface | addrobj]
```

其中 *address-type* 为 `dhcp` 或 `addrconf`。`addrconf` 参数指自动生成的 IPv6 地址。有关 *interface* 和 *addrobj* 选项的更详细说明，请参见上文中有关创建静态地址的说明。

#### 4. (可选) 显示有关新配置的 IP 接口的信息。

您可以根据要检查的信息使用以下命令：

```
# ipadm interface
```

如果未指定子命令，则将显示系统上所有接口的信息。

```
# ipadm show-if interface
```

如果未指定 *interface*，则会显示系统上所有接口的信息。

```
# ipadm show-addr interface|addrobj
```

如果未指定 *interface* 或 *addrobj*，则会显示所有地址对象的信息。

有关 `ipadm show-*` 子命令的输出的更多信息，请参见[“监视 IP 接口和地址” \[67\]](#)。

#### 5. 如果要配置使用主机名的静态 IP 地址，请在 `/etc/hosts` 文件中添加 IP 地址的对应条目。

此文件中的项由 IP 地址和其对应的主机名组成。

---

注 - 如果您要配置 DHCP 地址，则不需要更新 `/etc/hosts` 文件。

---

## 6. 定义缺省路由。

```
# route -p add default address
```

您可以使用 `netstat -r` 命令验证路由表的内容。

有关管理路由的更多信息，请参见 [route\(1M\)](#) 和“创建持久性（静态）路由” [54]。

### 例 3 为 IPv4 接口配置静态 IP 地址

以下示例说明了如何为接口配置静态 IP 地址。该示例首先在系统上启用 `DefaultFixed NCP` 以确保 `dladm` 和 `ipadm` 命令不修改反应性 NCP，这会使您执行的任何手动网络配置不起作用，具体取决于您的环境。

```
# netadm enable -p ncp DefaultFixed

# 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 192.168.84.3/24 net3
net3/v4

# 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
net3     ip         ok         --          --
net3/v4   static     ok         --          192.168.84.3/24

# vi /etc/hosts
# Internet host table
# 127.0.0.1    localhost
10.0.0.14    myhost
192.168.84.3 sales1

# route -p add default 192.168.84.1
# netstat -r
Routing Table: IPv4
Destination      Gateway          Flags Ref   Use      Interface
-----
default          192.168.84.1    UG    2    10466
192.168.84.0    192.168.84.3    U     3    1810    net0
```

```
localhost          localhost          UH    2    12    lo0

Routing Table: IPv6
Destination/Mask    Gateway            Flags Ref  Use  If
-----
solaris             solaris            UH    2    156  lo0
```

如果已在 `/etc/hosts` 文件中定义了 `sales1`，则在指定以下地址时可以使用该主机名：

```
# ipadm create-addr -a sales1 net3
net3/v4
```

#### 例 4 配置网络接口以从 DHCP 服务器接收 IP 地址

在以下示例中，IP 接口配置为从 DHCP 服务器接收其地址。DHCP 通常还会安装缺省路由。因此，该示例包括了手动添加缺省路由的步骤。

```
# 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
net3/v4

# ipadm
NAME    CLASS/TYPE    STATE    UNDER    ADDR
lo0     loopback     ok       --        --
lo0/v4  static       ok       --        127.0.0.1/8
net3    ip           ok       --        --
net3/v4 dhcp         ok       --        10.0.1.13/24
```

## 配置 IPv6 接口

作为在网络上使用 IPv6 寻址的初始步骤，必须在系统的 IP 接口上配置 IPv6。在安装过程中，可以在系统的一个或多个接口上启用 IPv6。

如果在安装期间启用 IPv6 支持，则在安装完成后，将存在以下与 IPv6 相关的文件和表：

- `name-service/switch` SMF 服务修改为包含使用 IPv6 地址的查找。
- 创建了 IPv6 地址选择策略表。该表确定通过启用了 IPv6 的接口进行传输时所用 IP 地址格式的优先级。

## ▼ 如何针对 IPv6 配置系统

以下过程说明了如何为安装 Oracle Solaris 之后添加的接口启用 IPv6。开始配置 IPv6 时，请首先在将成为 IPv6 节点的所有系统的接口上启用 IPv6。通常 IPv6 部署使用自动配置来配置 IP 接口。autoconf IP 地址指定链路本地地址并搜索在子网中使用的前缀和路由器。然后，您可以根据节点在 IPv6 网络中的作用（作为主机、服务器或路由器）来调整节点的配置。为 autoconf 设置的接口还将自动请求 DHCPv6 地址信息。要仅启用静态 IPv6 地址而不使用自动配置或 DHCPv6，请使用 `ipadm` 命令与适当的选项在接口上创建链路本地地址，而不添加任何其他动态指定的地址。有关示例，请参见“[从 IPv4 网络迁移到 IPv6 网络](#)” [51]。

---

注 - 如果接口与当前正在通告某个 IPv6 前缀的路由器在同一链路上，则接口会获取该站点前缀，并将其作为自动配置的地址的一部分。有关更多信息，请参见《[将 Oracle Solaris 11.2 系统配置为路由器或负载平衡器](#)》中的“[如何配置启用了 IPv6 的路由器](#)”。

---

1. (可选) 使用带有适当选项的 `ipadm create-ip` 命令配置 IP 接口。

```
# ipadm create-ip interface
```

例如，如下所示为 `net0` 配置 IP 接口：

```
# ipadm create-ip net0
```

如果已将接口配置为使用 IPv4，则不需要此步骤。有关配置 IP 接口的一般说明，请参见《[在 Oracle Solaris 11.2 中配置和管理网络组件](#)》中的“[如何配置 IPv4 接口](#)”。

2. 指定一个或多个 IP 地址。

---

注 - 指定 IP 地址时，请确保使用用于指定 IPv6 地址的正确选项：

```
# ipadm create-addr -T addrconf interface
```

要添加更多地址，请使用以下语法：

```
# ipadm create-addr -a ipv6-address interface
```

3. (可选) 创建静态 IPv6 缺省路由。

```
# /usr/sbin/route -p add -inet6 default ipv6-address
```

---

注 - 作为自动配置过程的一部分，`in.ndpd` 在搜索到缺省路由时会添加这些路由，这可能导致有多个缺省路由可用，包括任何手动配置的缺省路由。系统根据所有可用路由自动做出缺省路由选择，这意味着不会每次都使用手动配置的缺省路由。

---

4. (可选) 创建一个 `/etc/inet/ndpd.conf` 文件，以定义节点上的接口变量的参数。

如果需要为主机的接口创建临时地址，请参见“为 IPv6 接口使用临时地址” [46]。有关 `/etc/inet/ndpd.conf` 的更多信息，请参见 `ndpd.conf(4)` 手册页。

5. (可选) 显示 IP 接口的状态及其 IPv6 配置，如下所示：

```
# ipadm show-addr
```

例 5 在安装之后启用 IPv6 接口

以下示例说明了如何在 `net0` 接口上启用 IPv6。在开始之前，请检查系统上已配置的所有接口的状态。

```
# 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
```

如以上输出中所示，目前仅为该系统配置了 `net0` 接口。如果尚未配置 `net0` 接口，请使用 `ipadm create-ip net0` 命令来初启接口。

然后，如下所示在此接口上启用 IPv6：

```
# ipadm create-addr -T addrconf net0
# ipadm create-addr -a 2001:db8:3c4d:15::203/64 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     fe80::203:baff:fe13:14e1/10
lo0/v6       static   ok     ::1/128
net0/v6a     static   ok     2001:db8:3c4d:15::203/64

# route -p add -inet6 default fe80::203:baff:fe13:14e1
```

- 接下来的步骤
- 有关如何将 IPv6 节点配置为路由器的信息，请参见《将 Oracle Solaris 11.2 系统配置为路由器或负载均衡器》中的“配置 IPv6 路由器”。
  - 有关如何将节点定制为服务器的信息，请参见“在服务器上配置启用了 IPv6 的接口” [51]。

## 为 IPv6 接口使用临时地址

IPv6 临时地址包括一个随机生成的用作接口 ID 的 64 位数字，而不是包括接口的 MAC 地址。对于要保持匿名的 IPv6 节点上的任何接口都可以使用临时地址。例如，您可能希望对于需要访问公共 Web 服务器的主机的接口使用临时地址。临时地址可实现 IPv6 保密性增强功能。RFC 3041 中介绍了这些增强功能，可从《Privacy Extensions for

Stateless Address Autoconfiguration in IPv6》（《IPv6 中用于无状态地址自动配置的专用扩展》）(<http://www.rfc-editor.org/rfc/rfc3041.txt>)中获取。

如果需要的话，可以在 `/etc/inet/ndpd.conf` 文件中为一个或多个接口启用临时地址。但是，与自动配置的标准 IPv6 地址不同，临时地址由 64 位子网前缀和一个随机生成的 64 位数字组成。这个随机数将成为 IPv6 地址的接口 ID 部分。临时地址作为接口 ID 时，不会生成链路本地地址。

请注意，临时地址的缺省首选生命周期为一天。启用临时地址生成功能时，还可以在 `/etc/inet/ndpd.conf` 文件中配置下列变量：

<code>valid lifetime</code> <code>TmpValidLifetime</code>	临时地址存在的时间跨度，在此之后临时地址将从主机中删除。
<code>preferred lifetime</code> <code>TmpPreferredLifetime</code>	临时地址过时之前已经过的时间。此时间跨度应短于有效生命周期。
<code>address regeneration</code>	在首选生命周期到期之前的持续时间，在这段时间内，主机应生成新的临时地址。

可以按如下所示表示临时地址的持续时间：

<code>n</code>	<code>n</code> 秒数（缺省值）
<code>n h</code>	<code>n</code> 小时数 (h)
<code>n d</code>	<code>n</code> 天数 (d)

## ▼ 如何配置临时 IPv6 地址

- 如有必要，请在主机的接口上启用 IPv6。  
请参阅[如何针对 IPv6 配置系统 \[45\]](#)。
- 编辑 `/etc/inet/ndpd.conf` 文件以开启临时地址生成功能。
  - 要在主机的所有接口上配置临时地址，请将以下行添加到 `/etc/inet/ndpd.conf` 文件中：
 

```
ifdefault TmpAddrsEnabled true
```
  - 要配置特定接口的临时地址，请将以下行添加到 `/etc/inet/ndpd.conf` 文件中：
 

```
if interface TmpAddrsEnabled true
```
- （可选）指定临时地址的有效生命周期。

```
ifdefault TmpValidLifetime duration
```

此语法为主机上的所有接口指定有效生命周期。*duration* 的值应当以秒、小时或天为单位。缺省的有效生命周期为 7 天。另外，还可以将 `TmpValidLifetime` 与 `if interface` 关键字一起使用来为特定接口的临时地址指定有效生命周期。

4. (可选) 为临时地址指定所需的生命周期，在此时间之后临时地址将过时。

```
if interface TmpPreferredLifetime duration
```

此语法为特定接口的临时地址指定首选生命周期。缺省的首选生命周期为一天。另外，还可以将 `TmpPreferredLifetime` 与 `ifdefault` 关键字一起使用来为主机所有接口上的临时地址指定所需的生命周期。

---

注 - 缺省地址选择可为已经过时的 IPv6 地址指定较低的优先级。如果某个 IPv6 临时地址已过时，则缺省地址选择会选择了一个未过时的地址作为包的源地址。未过时的地址可能是自动生成的 IPv6 地址，也可能是接口的 IPv4 地址。有关缺省地址选择的更多信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的“管理缺省地址选择”。

---

5. (可选) 指定地址过时之前的交付周期，在这段时间内，主机应生成新的临时地址。

```
ifdefault TmpRegenAdvance duration
```

此语法可为主机上所有接口的临时地址指定地址过时之前的交付周期。缺省值是 5 秒。

6. 如下所示更改 `in.ndpd` 守护进程的配置：

```
# pkill -HUP in.ndpd
# /usr/lib/inet/in.ndpd
```

7. 通过发出 `ipadm show-addr` 命令验证是否已创建了临时地址，如例 6 “显示启用了临时地址的 `ipadm show-addr` 命令输出”中所述。

该命令输出显示临时地址的 `CURRENT` 字段中的 `t` 标志。

#### 例 6 显示启用了临时地址的 `ipadm show-addr` 命令输出

以下示例显示了创建临时地址之后 `ipadm show-addr` 命令的输出。请注意，样例输出中仅包括 IPv6 相关信息。

```
# ipadm show-addr -o all
ADDROBJ  TYPE    STATE CURRENT PERSISTENT ADDR
lo0/v6   static  ok     U----   ---      ::1/128
net0/v6   addrconf ok     U----   ---      fe80::a00:20ff:feb9:4c54/10
net0/v6a  static  ok     U----   ---      2001:db8:3c4d:15:a00:20ff:feb9:4c54/64
net0/?   addrconf ok     U--t-   ---      2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64
```

请注意，对于地址对象 `net0/?`，在 `CURRENT` 字段下设置了 `t` 标记，指示对应的地址具有临时接口 ID。

- 另请参见
- 要设置对 IPv6 地址的名称服务支持，请参见第 4 章 在 Oracle Solaris 客户机上管理命名和目录服务。
  - 要为服务器配置 IPv6 地址，请参见[如何配置用户指定的 IPv6 标记 \[49\]](#)。
  - 要监视 IPv6 节点上的活动，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的第 1 章, “管理 TCP/IP 网络”。

## 配置 IPv6 标记

IPv6 地址的 64 位接口 ID 也称为标记。在地址自动配置过程中，该标记与接口的 MAC 地址相关联。大多数情况下，非路由节点（IPv6 主机和服务器）应当使用为其自动配置的标记。

但是，对于具有在系统维护期间需要例行交换的接口的服务器，使用自动配置的标记可能会产生问题。如果更换接口卡，则 MAC 地址也会随之更改。因此，依赖稳定 IP 地址的服务器将会遇到问题。网络基础结构的各个部分，例如域名系统 (Domain Name System, DNS) 或网络信息系统 (Network Information System, NIS) 可能已为服务器的接口存储了特定 IPv6 地址。

为了避免出现地址更改问题，可以手动配置要用作 IPv6 地址中接口 ID 的标记。要创建此标记，需要指定一个 64 位或更少的十六进制数字，使其占用 IPv6 地址的接口 ID 部分。在后续的地址自动配置过程中，相邻节点搜索协议不会基于接口的 MAC 地址创建接口 ID。相反，手动创建的标记将成为接口 ID。此标记始终被指定给该接口，即使更换了卡也是如此。

---

注 - 用户指定的标记和临时地址之间的区别在于，临时地址是随机生成的，而不是由用户显式创建的。

---

### ▼ 如何配置用户指定的 IPv6 标记

对于具有需要例行交换的接口的服务器，以下过程尤其有用。还可以按照以下步骤在任意 IPv6 节点上配置用户指定的标记。

1. 确认要为其配置标记的接口存在并且未在该接口上配置 IPv6 地址。

```
# 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
```

以上输出显示存在未配置 IPv6 地址的网络接口 net0。

2. 创建一个或多个要用作节点接口标记的 64 位十六进制数字，采用以下格式：

```
XXXX:XXXX:XXXX:XXXX
```

3. 配置将具有用户指定的接口 ID (标记) 的各个接口。

```
# ipadm create-addr -T addrconf -i interface-ID interface
```

例如，可以如下所示配置具有标记的接口 net0：

```
# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
```

---

注 - 为地址对象创建标记之后，无法再修改该标记。

---

4. 使用所做更改更新 IPv6 守护进程。

```
# pkill -HUP in.ndpd
```

#### 例 7 在 IPv6 接口上配置用户指定的标记

以下示例说明了如何为 net0 配置 IPv6 地址和标记。

```
# 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
# pkill -HUP in.ndpd
# ipadm show-addr
ADDROBJ TYPE STATE ADDR
lo0/v6 static ok ::1/128
net0/v6 addrconf ok fe80::1a:2b:3c:4d/10
net0/v6a addrconf ok 2002:a08:39f0:1:1a:2b:3c:4d/64
```

在配置标记之后，地址对象 net0/v6 同时具有一个链路本地地址和一个为其接口 ID 配置了 1a:2b:3c:4d 的地址。请注意，创建 net0/v6 之后，无法再为该接口修改此标记。

- 另请参见
- 要使用服务器的 IPv6 地址更新名称服务，请参见第 4 章在 Oracle Solaris 客户机上管理命名和目录服务。
  - 要监视服务器性能，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的第 1 章，“管理 TCP/IP 网络”。

## 在服务器上配置启用了 IPv6 的接口

在服务器上针对 IPv6 地址进行规划时，当在服务器的接口上启用 IPv6 时，必须做出几个决定。所做的决定会影响用于配置接口 IPv6 地址的接口 ID（标记）的策略。

### ▼ 如何在服务器接口上启用 IPv6

以下过程说明了如何在网络服务器上启用 IPv6。某些步骤随您希望用来实施 IPv6 的方式不同可能有所不同。

1. 在服务器的 IP 接口上启用 IPv6。  
有关逐步说明，请参见[“配置 IPv6 接口” \[44\]](#)。
2. 确保在与服务器位于同一链路的路由器上配置了 IPv6 子网前缀。  
有关更多信息，请参见《[将 Oracle Solaris 11.2 系统配置为路由器或负载均衡器](#)》中的[“配置 IPv6 路由器”](#)。
3. 选择以下策略之一，以便将接口 ID 指定给服务器上启用了 IPv6 的接口。  
缺省情况下，在创建 IPv6 地址的接口 ID 部分时，IPv6 地址自动配置会使用接口的 MAC 地址。如果接口的 IPv6 地址是已知的，则使用一个接口交换另一个接口会导致问题。新接口的 MAC 地址将会不同。在地址自动配置过程中，会生成新的接口 ID。
  - 对于您未计划更换的已启用 IPv6 的接口，请使用自动配置的 IPv6 地址。
  - 对于在本地网络外部必须匿名显示的启用了 IPv6 的接口，请考虑为接口 ID 使用随机生成的标记。  
有关更多信息，请参见[如何配置临时 IPv6 地址 \[47\]](#)
  - 对于计划定期交换的启用了 IPv6 的接口，可以使用静态配置，也可以为接口 ID 创建标记。  
有关更多信息，请参见[如何配置用户指定的 IPv6 标记 \[49\]](#)。

## 从 IPv4 网络迁移到 IPv6 网络

---

注 - 从 IPv4 网络迁移到 IPv6 网络之前，请注意大部分迁移计划涉及到在相当长的时间内同时运行 IPv4 和 IPv6，并且可能必须一直如此。

---

在从 IPv4 网络迁移到 IPv6 网络之前，请检查《[在 Oracle Solaris 11.2 中规划网络部署](#)》中的第 2 章，“[规划 IPv6 地址的使用](#)”中的信息，确定是否需要执行任何附加任务。

从 IPv4 网络迁移到 IPv6 网络的基本步骤涉及到首先删除所有现有 IPv4 DHCP 和静态 IP 地址，然后根据需要重新配置尽可能多的新 IPv6 地址。如果新 IPv6 接口与当前通告了某个 IPv6 前缀的路由器位于同一链路上，则接口会获取链路前缀。有关更多信息，请参见《[将 Oracle Solaris 11.2 系统配置为路由器或负载均衡器](#)》中的“[配置 IPv6 路由器](#)”。

例 8 将 IPv4 地址迁移到 IPv6 地址

以下示例说明了如何将现有 IPv4 地址迁移到 IPv6 地址。该过程首先删除所有现有 IPv4 DHCP 和静态 IP 地址。

```
# ipadm show-addr net0/  
ADDROBJ  TYPE  STATE  ADDR  
lo0/v4   static ok     127.0.0.1/8  
net0/v4  static ok     172.16.27.74/24  
# ipadm delete-addr net0/v4
```

有关说明，请参见“[删除或修改 IP 接口配置](#)” [66]。

接下来，使用带有适当选项和参数的 ipadm create addr 命令创建新的 IPv6 地址。

例如，可以如下所示创建链路本地地址和 addrconf IPv6 地址：

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

如下所示创建不使用 DHCPv6 的静态 IPv6 地址和 addrconf 地址：

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

如下所示创建静态 IPv6 地址：

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

使用 ipadm show-addr 命令显示新的 IPv6 配置。

有关本例中未包括的其他 IPv6 配置步骤（必需和可选），请参见“[配置 IPv6 接口](#)” [44]。

## 配置路由

本节包含以下主题：

- “[路由表和路由类型](#)” [53]
- “[创建持久性（静态）路由](#)” [54]
- “[为单接口系统启用路由](#)” [56]

## 路由表和路由类型

路由器和主机都维护着一个路由表。例如，以下路由表列出了系统知晓的网络的 IP 地址，包括系统的本地缺省网络的 IP 地址。该表还列出了每个已知网络的网关系统的 IP 地址。网关是一个系统，它可以接收传出包并将它们转发到距本地网络一个跃点的位置。

```
Routing Table: IPv4
Destination          Gateway             Flags Ref  Use  Interface
-----
default              172.20.1.10        UG     1    532  net0
224.0.0.0            10.0.5.100         U      1     0    net1
10.0.0.0             10.0.5.100         U      1     0    net1
127.0.0.1            127.0.0.1          UH     1     57   lo0
```

可以在 Oracle Solaris 系统上配置以下两种类型的路由：静态路由和动态路由。可以在单个系统上配置其中一种或两种路由类型。实现动态路由的系统依赖路由协议（例如用于 IPv4 网络的路由信息协议 (Routing Information Protocol, RIP) 和用于 IPv6 网络的 RIPng (RIP next generation)），对网络通信进行路由并更新表中的路由信息。对于静态路由，信息是通过使用 `route` 命令手动维护的。有关更多信息，请参见 [route\(1M\)](#) 手册页。

为本地网络或自治系统 (autonomous system, AS) 配置路由时，请考虑要在特定的路由器和主机上支持哪种类型的路由。下表显示了各种路由类型，以及各个路由类型分别最适用于哪种网络方案。

路由类型	最佳用途
静态	从缺省路由器获取其路由的小型网络和主机，以及仅需要知晓接下来几个跃点上的一个或两个路由器的缺省路由器。
动态	较大的互联网络，包括具有许多主机的本地网络中的路由器以及大型自治系统上的主机。动态路由是适用于大多数网络中的系统的最佳选项。
组合的静态和动态路由	将静态路由网络和动态路由网络连接在一起的路由器，以及将内部自治系统与外部网络连接在一起的边界路由器。将系统上的静态路由和动态路由组合在一起是一种常见的做法。

《在 Oracle Solaris 11.2 中规划网络部署》中的“IPv4 自治系统拓扑”所述的拓扑中组合使用了静态路由和动态路由。

---

注 - 指向同一目的地的两个路由不会自动导致系统执行负载平衡或故障转移。如果需要这些功能，请使用 IPMP。有关更多信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的第 2 章，“关于 IPMP 管理”。

---

## 创建持久性（静态）路由

您可以使用 `route` 命令手动处理网络路由表。要使更改在重新引导之后仍然保留，请使用 `-p` 选项。由于 `/etc/defaultrouter` 文件在 Oracle Solaris 11 中已过时，因此您不能再使用此文件管理路由（缺省值或其他）。使用 `route` 命令是唯一一种手动使路由在系统重新引导之后仍然保留的方法。

---

注 - `route` 命令仅处理活动配置文件的路由。如果活动配置文件发生更改，则缺省路由以及所有其他路由可能会被替换。但是，如果始终在系统上使用相同的配置文件，则不需要考虑这个问题。

---

添加持久性路由时，请务必确保持久性配置中尚不存在所添加的路由。如果持久性配置中已存在这些路由，则网络路由表会更改，但不更新该持久性路由。这种情况的一个例子是系统的缺省路由映射到了系统的主接口（在安装 Oracle Solaris 之后经常会出现这种情况）。如果接下来将系统的主接口更改为其他接口，则系统的缺省路由也应执行持久性更新。最佳做法是在添加新路由之前删除持久性路由配置。有关更多信息，请参见《在 Oracle Solaris 11.2 中排除网络管理问题》中的“排除添加持久路由时的问题”。

请注意有关创建和显示持久性路由的以下附加信息：

- 结合使用 `route` 命令与 `-p` 选项添加持久性路由：

```
# route -p add default ip-address
```

对于使用此方法创建的路由，可使用 `route -p show` 命令来显示所有持久性静态路由：

```
# route -p show
```

- 结合使用 `netstat` 命令与以下选项，可以显示系统上当前处于活动状态的路由：

```
# netstat -rn
```

请参见 [netstat\(1M\)](#) 和 [route\(1M\)](#) 手册页。

有关更多信息，请参见 [netstat\(1M\)](#) 和 [route\(1M\)](#) 手册页。

有关在使用反应性模式时创建和显示缺省路由的信息，请参见第 5 章 [关于在 Oracle Solaris 中管理基于配置文件的网络配置](#)。

### ▼ 如何将静态路由添加到路由表

1. 使用常规用户帐户查看路由表的当前状态。

```
% netstat -rn
```

输出将类似于以下内容：

```

Routing Table: IPv4
  Destination      Gateway           Flags Ref    Use    Interface
-----
192.168.5.125     192.168.5.10    U      1     5879   net0
224.0.0.0         198.168.5.10    U      1      0     net0
default           192.168.5.10    UG     1    91908
127.0.0.1         127.0.0.1       UH     1    811302  lo0

Routing Table: IPv6
  Destination/Mask  Gateway           Flags Ref    Use    If
-----
::1                ::1               UH     2      0     lo0
    
```

2. 成为管理员。
3. (可选) 刷新路由表中的现有条目。

```
# route flush
```

4. 添加持久性路由。

```
# route -p add -net network-address -gateway gateway-address
```

-p 创建一个在系统重新引导后仍然保留的路由。如果希望路由仅在当前会话中保留，则不要使用 -p 选项。

-net *network-address* 指定路由将转到具有 *network-address* 中指定的地址的网络。

-gateway *gateway-address* 指示指定路由的网关系统具有 IP 地址 *gateway-address*。

#### 例 9 将静态路由添加到路由表

以下示例说明了如何将静态路由添加到路由器（路由器 2）。AS 的边界路由器 10.0.5.150 需要静态路由。有关此特定设置的说明，请参见图 2。

如下所示查看路由器 2 上的路由表：

```

# netstat -rn
Routing Table: IPv4
  Destination      Gateway           Flags Ref    Use    Interface
-----
default           172.20.1.10      UG     1     249   ce0
224.0.0.0         172.20.1.10      U      1      0     ce0
10.0.5.0          10.0.5.20        U      1     78    bge0
127.0.0.1         127.0.0.1        UH     1     57    lo0

Routing Table: IPv6
    
```

Destination/Mask	Gateway	Flags	Ref	Use	If
::1	::1	UH	2	0	lo0

路由表指示路由器 2 知晓两个路由。缺省路由将路由器 2 的 172.20.1.10 接口用作其网关。第二个路由 10.0.5.0 是由在路由器 2 上运行的 in.routed 守护进程搜索到的。此路由的网关是路由器 1，它的 IP 地址是 10.0.5.20。

如下所示将第二个路由添加到网络 10.0.5.0，该网络使用其网关作为边界路由器：

```
# route -p add -net 10.0.5.0/24 -gateway 10.0.5.150
add net 10.0.5.0: gateway 10.0.5.150
```

现在，路由表中具有边界路由器（其 IP 地址为 10.0.5.150）的一个路由。

```
# netstat -rn
Routing Table: IPv4
Destination          Gateway             Flags  Ref  Use  Interface
-----
default              172.20.1.10        UG     1    249 ce0
224.0.0.0            172.20.1.10        U      1     0 ce0
10.0.5.0             10.0.5.20          U      1     78 bge0
10.0.5.0             10.0.5.150         U      1    375 bge0
127.0.0.1            127.0.0.1          UH     1     57 lo0

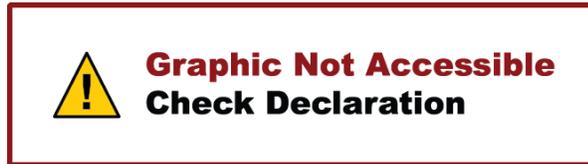
Routing Table: IPv6
Destination/Mask     Gateway             Flags  Ref  Use  If
-----
::1                  ::1                 UH     2     0 lo0
```

## 为单接口系统启用路由

可以为单接口系统配置静态或动态路由。使用静态路由时，主机必须依赖于缺省路由器的服务来获取路由信息。启用使用某种路由协议的动态路由是管理系统上的路由的最简便方法。

具有多个路由器和网络的站点通常将其网络拓扑作为单个路由域或自治系统 (autonomous system, AS) 进行管理。本节中的过程和示例以下图为基础。在该图中，AS 划分为三个本地网络：10.0.5.0、172.20.1.0 和 192.168.5.0。网络由路由器和客户机系统组成，包括以下类型的路由器：边界路由器、缺省路由器和包转发路由器。客户机系统包括多宿主系统和单接口系统。有关其中各个组件的更多详细信息，请参见《在 Oracle Solaris 11.2 中规划网络部署》中的“IPv4 自治系统拓扑”。

图 2 具有多个 IPv4 路由器的自治系统



## ▼ 如何在单接口系统上启用动态路由

以下过程假定您已配置了系统的 IP 接口。有关针对网络上的路由器进行规划的更多详细信息，请参见《在 Oracle Solaris 11.2 中规划网络部署》中的“IPv4 自治系统拓扑”。

1. 成为管理员。
2. 为系统的其中一个 IP 接口配置系统所属网络的 IP 地址。  
有关说明，请参见[如何配置 IPv4 接口 \[40\]](#)。
3. 从系统中删除所有持久定义的路由器。  
执行此步骤是因为存在任何静态定义的缺省路由会阻止系统在引导期间启用动态路由。

- a. 如下所示确定所有持久定义的缺省路由：

```
# route -p show
```

- b. 删除各个持久定义的路由。例如：

```
# route -p delete -net default -gateway 172.20.1.10
```

4. 确保包转发已禁用。
5. 在系统上启用 IPv4 路由。

```
# routeadm -e ipv4-routing -u
```

### 例 10 在单接口系统上运行动态路由

以下示例说明了如何为 hosta（网络 192.168.5.0 上的一个单接口系统）配置动态路由，如[图 2](#)中所示。该系统将路由器 1 用作其缺省路由器。此示例假定您已配置了系统的 IP 接口。

首先，您应使用管理员权限登录到 `hosta`。然后，从系统中删除所有持久定义的路由。

```
# route -p show
persistent: route add default 172.20.1.10

# route -p delete default 172.20.1.10
delete net default: gateway 172.20.1.10
delete persistent net default: gateway 172.20.1.10

# routeadm
Configuration      Current           Current
      Option      Configuration      System State
-----
      IPv4 routing disabled          disabled
      IPv6 routing disabled          disabled
      IPv4 forwarding disabled        disabled
      IPv6 forwarding disabled        disabled

Routing services   "route:default ripng:default"

Routing daemons:

STATE  FMRI
disabled svc:/network/routing/ripng:default
online  svc:/network/routing/ndp:default
disabled svc:/network/routing/rdisc:default
disabled svc:/network/routing/legacy-routing:ipv4
disabled svc:/network/routing/legacy-routing:ipv6
disabled svc:/network/routing/route:default

# routeadm -d ipv4-forwarding -u
# routeadm -e ipv4-routing -u
# routeadm
Configuration      Current           Current
      Option      Configuration      System State
-----
      IPv4 routing enabled          enabled
      IPv6 routing disabled          disabled
      IPv4 forwarding disabled        disabled
      IPv6 forwarding disabled        disabled

Routing services   "route:default ripng:default"

Routing daemons:

STATE  FMRI
disabled svc:/network/routing/ripng:default
online  svc:/network/routing/ndp:default
disabled svc:/network/routing/rdisc:default
disabled svc:/network/routing/legacy-routing:ipv4
disabled svc:/network/routing/legacy-routing:ipv6
online  svc:/network/routing/route:default
```

## 关于 IPv6 路由

在 CIDR 下，IPv6 路由与 IPv4 路由几乎完全相同。唯一的区别在于地址是 128 位 IPv6 地址，而非 32 位 IPv4 地址。借助非常直接的扩展，所有 IPv4 的路由算法，例如开放最短路径优先 (Open Shortest Path First, OSPF)、RIP、域间路由协议 (Inter-domain Routing Protocol, IDRP) 和中间系统到中间系统 (Intermediate System to Intermediate System, IS-IS) 可用于对 IPv6 进行路由。

IPv6 还包括下列支持功能强大的新路由功能的简单路由扩展：

- 基于策略、性能、成本等因素选择提供商
- 主机灵活性，可路由到当前位置
- 自动重新寻址，可路由到新地址

通过创建可使用 IPv6 路由选项的 IPv6 地址序列，可以获取新路由功能。IPv6 源使用路由选项列出在前往包目的地的途中要访问的中间节点（一个或多个）或拓扑组。此功能与 IPv4 的稀疏源路由选项和记录路由选项非常相似。

在大多数情况下，为了使地址序列成为一般功能，必须使用 IPv6 主机将主机接收的包中的路由反向。包必须使用 IPv6 验证头成功地进行验证。包中必须包含地址序列才能将包返回到其始发者。此方法会强制所实现的 IPv6 主机支持对源路由进行处理和反向。源路由的处理和反向使提供商能够使用实现了新 IPv6 功能（例如提供程序选择和扩展地址）的主机。

## 配置多宿主主机

在 Oracle Solaris 中，具有多个接口的系统被视为多宿主主机。多宿主主机的各个接口可以连接到不同物理网络或同一物理网络上的不同子网。有关创建多宿主主机的逐步说明，请参见[如何创建多宿主主机 \[60\]](#)。

如果一个系统具有多个接口连接到同一子网，必须首先将这些接口配置到一个 IPMP 组中。否则，该系统无法成为多宿主主机。有关 IPMP 的更多信息，请参见《[在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道](#)》中的第 2 章，“关于 IPMP 管理”。

多宿主主机不会转发 IP 包，但可以将多宿主主机配置为运行路由协议。通常，可以将以下类型的系统配置为多宿主主机：

- 可以将 NFS 服务器（尤其是用作大型数据中心的那些服务器）连接到多个网络，以便在大量用户之间共享文件。这些服务器无需维护路由表。
- 数据库服务器可以具有多个网络接口，从而可为大量用户提供资源，就像 NFS 服务器那样。
- 防火墙网关是连接公司网络和公共网络（如 Internet）的系统。管理员将设置防火墙作为一项安全措施。当配置为防火墙时，主机不在连接到主机接口的网络之间传递包。但是，主机仍可以向经授权的用户提供标准 TCP/IP 服务，如 ssh。

---

注 - 当多宿主主机在其任一接口上具有不同类型的防火墙时，请小心谨慎以免无意中中断主机的包。对于有状态防火墙，尤其容易出现此问题。针对此问题的一种解决方案是配置无状态防火墙。有关防火墙的更多信息，请参见《在 Oracle Solaris 11.2 中确保系统和连接设备的安全》中的“防火墙系统”或第三方防火墙的相应文档。

---

## ▼ 如何创建多宿主主机

1. 成为管理员。
2. 配置在安装过程中没有配置的每个其他网络接口。  
请参见[如何配置 IPv4 接口 \[40\]](#)。
3. 如果启用了包转发，请将此服务禁用。

```
# routeadm -p ipv4-forwarding
persistent=enabled default=disabled current=enabled
# routeadm -d ipv4-forwarding -u
# routeadm -p ipv4-forwarding
persistent=disabled default=disabled current=disabled
```

4. (可选) 为多宿主主机开启动态路由。

```
# routeadm -e ipv4-routing -u
# routeadm -p ipv4-routing
persistent=enabled default=enabled current=enabled
```

### 例 11 配置多宿主主机

以下示例说明了如何配置多宿主主机，如《在 Oracle Solaris 11.2 中规划网络部署》中的“IPv4 自治系统拓扑”中的图所示。在该示例中，系统具有主机名 `hostc`。此主机具有两个接口，这两个接口都已连接到网络 192.168.5.0。

要开始操作，请显示系统接口的状态。

```
# dladm show-link
LINK      CLASS      MTU      STATE    BRIDGE    OVER
net0      phys       1500     up       --        --
net1      phys       1500     up       --        --

# ipadm show-addr
ADDROBJ   TYPE      STATE    ADDR
lo0/v4    static   ok       127.0.0.1/8
net0/v4    static   ok       192.168.5.82/24
```

dladm show-link 命令报告 hostc 具有两个数据链路。但是，只有 net0 配置有 IP 地址。要将 hostc 配置为多宿主主机，应当为 net1 配置同一 192.168.5.0 网络中的一个 IP 地址。确保 net1 的底层物理 NIC 已通过物理方式连接到网络。

```
# ipadm create-ip net1
# ipadm create-addr static -a 192.168.5.85/24 net1
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
net0/v4       static    ok         192.168.5.82/24
net1/v4       static    ok         192.168.5.85/24
```

接下来，如下所示将 net1 接口添加到 /etc/hosts 文件：

```
# vi /etc/inet/hosts
127.0.0.1      localhost
192.168.5.82   hostc   #primary network interface for host3
192.168.5.85   hostc-2 #second interface
```

然后，如下所示关闭包转发（如果此服务正在 hostc 上运行）：

```
# routeadm -p ipv4-forwarding
persistent=enabled default=disabled current=enabled

# routeadm
Configuration      Current      Current
Option              Configuration System State
-----
IPv4 routing        enabled      enabled
IPv6 routing        disabled     disabled
IPv4 forwarding     disabled     disabled
IPv6 forwarding     disabled     disabled

Routing services    "route:default ripng:default"

Routing daemons:

STATE  FMRI
disabled svc:/network/routing/ripng:default
online  svc:/network/routing/ndp:default
disabled svc:/network/routing/rdisc:default
disabled svc:/network/routing/legacy-routing:ipv4
disabled svc:/network/routing/legacy-routing:ipv6
online  svc:/network/routing/route:default
```

routeadm 命令报告当前启用了通过 in.routed 守护进程的动态路由。

## 在多宿主主机上实现对称路由

缺省情况下，具有多个接口的系统（也称为多宿主主机）基于路由表中到通信目的地的最长匹配路由来路由其网络通信。如果存在多个长度相等的到目的地的路由路径，则 Oracle Solaris 将应用等成本多路径 (Equal-Cost Multi-Path, ECMP) 算法将通信分布到这些路由。

以此方式对通信进行分布并非始终合适。例如，可能会通过多宿主主机上的一个接口发送 IP 包，但是此主机与该包中的 IP 源地址不位于同一子网上。此外，如果传出数据包是对某传入请求（例如 ICMP 回显请求）的响应，则请求和响应可能不经过相同的接口。此类型的通信路由配置称为非对称路由。如果您的 Internet 服务提供商 (Internet service provider, ISP) 按照 RFC 3704 (<http://www.rfc-editor.org/rfc/bcp/bcp84.txt>) 中所述实现准入过滤，则非对称路由配置可能导致 ISP 丢弃传出数据包。

RFC 3704 旨在限制拒绝服务 (denial-of-service, DoS) 攻击在 Internet 上扩散。为了与它保持一致，必须将您的网络配置为对称路由。使用 IP `hostmodel` 属性可以满足此要求。此属性控制通过多宿主主机接收或传送 IP 数据包的行为。

`hostmodel` 属性可以具有以下三个可能值之一：

<code>strong</code>	对应于 RFC 1122 中定义的强端系统 (end system, ES) 模型。此值实现对称路由。
<code>weak</code>	对应于 RFC 1122 中定义的弱 ES 模型。在采用此值的情况下，多宿主主机使用非对称路由。
<code>src-priority</code>	通过使用首选路由配置数据包路由。如果路由表中存在多个目的地路由，则首选路由是使用在其上配置了传出数据包 IP 源地址的接口的那些路由。如果不存在这样的路由，则传出数据包将使用到数据包的目的地的最长匹配路由。

例如，如下所示在多宿主主机上实现 IP 包的对称路由：

```
# ipadm set-prop -p hostmodel=strong ipv4
# ipadm set-prop -p hostmodel=strong ipv6
# ipadm show-prop -p hostmodel ip
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
ipv6 hostmodel rw strong -- weak strong,
src-priority,
weak
ipv4 hostmodel rw strong -- weak strong,
src-priority,
weak
```

## 定制 IP 接口属性和地址

有三个 `ipadm` 子命令可用于管理 IP 接口属性：

- `show-ifprop -p property interface` – 显示 IP 接口的属性及其当前值。如果不使用 `-p property` 选项，则会列出 IP 接口的所有属性。如果没有指定 IP 接口，则会列出所有 IP 接口的所有属性。
- `set-ifprop -p property=value interface` – 为 IP 接口的属性赋值。
- `reset-ifprop -p property interface` – 将特定属性重置为其缺省值。

与数据链路一样，IP 接口还具有您可以为特定网络环境定制的属性。对于每个接口，存在两组属性，一组适用于 IPv4，另一组适用于 IPv6。

## 设置 MTU 属性

数据链路和 IP 接口都具有某些属性（包括 MTU 属性）。因此，您可以有一个针对数据链路的 MTU 值和另一个针对在该链路上配置的接口的 MTU 值。而且，在该 IP 接口上，您可以分别为穿越该 IP 接口的 IPv4 和 IPv6 数据包应用不同的 MTU 值。

为 IP 接口设置 MTU 属性时，请记住以下要点：

- IP 接口的 MTU 设置值不能大于数据链路的 MTU 设置值。在这种情况下，`ipadm` 命令会显示错误消息。
- 如果某个 IP 接口的 MTU 值不同于数据链路的 MTU 值，则 IP 包将限制为该 IP 接口的 MTU 值。例如，如果数据链路的 MTU 值为 9000 字节，而 IP 接口的 MTU 值为 1500 字节，则 IP 包将限制为 1500 字节。不过，使用底层第 2 层协议的其他第 3 层协议可以发送最多 9000 字节的包。

有关定制数据链路属性的说明，包括数据链路 MTU 设置如何影响 IP 接口的 MTU 设置，请参见[“定制数据链路属性” \[29\]](#)。

## 启用包转发

在网络上，一个主机可以接收发送到另一个主机系统的数据包。通过在作为接收方的本地系统上启用包转发，该系统可以将数据包转发到目的地主机。此过程称为 *IP* 转发，缺省情况下在 Oracle Solaris 中被禁用。

包转发功能由可同时在 IP 接口和 TCP/IP 协议上设置的一个属性予以管理。如果希望可以选择包转发方式，则可以在 IP 接口上启用包转发。例如，您的系统可能有多个 NIC，其中某些 NIC 连接到外部网络，另一些 NIC 连接到专用网络。因此，您将只对部分接口（而不是所有接口）启用包转发。

通过设置 TCP/IP 协议的属性，还可以全局性地在系统中启用包转发。有关更多信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的“全局启用包转发”。

注 - IP 接口的 forwarding 属性与协议的该属性不是互斥的。您可以同时为接口和协议设置该属性。例如，可以在协议上全局性地启用包转发，然后在系统上定制每个 IP 接口的包转发。因此，尽管全局启用包转发，但系统仍可选择包转发。

例如，如下所示在 IP 接口上启用包转发：

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

其中 *protocol-version* 为 IPv4 或 IPv6。必须分别针对 IPv4 和 IPv6 包键入此命令。

以下示例说明了如何在系统上仅启用 IPv4 包转发：

```
# 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
...
```

## 定制 IP 地址属性

可以使用 `ipadm` 命令管理特定于 IP 地址的属性。

您可以定制 IP 地址属性以管理以下网络配置参数：

- 网络掩码长度
- IP 地址是否可用作传出数据包的源地址
- 地址是属于全局区域还是非全局区域
- 地址是否为专用地址

处理 IP 地址属性时可以使用以下 `ipadm` 子命令：

- `show-addrprop -p property addrobj` - 显示地址属性，具体取决于所用选项。  
要显示所有 IP 地址的属性，请勿指定属性或地址对象。要显示所有 IP 地址的单个属性的值，请仅指定该属性。要显示特定地址对象的所有属性，请仅指定该地址对象。
- `set-addrprop -p property=value addrobj` - 为地址属性赋值。请注意，一次只能设置一个地址属性。

- `reset-addrprop -p property addrobj` – 恢复地址属性的任意缺省值。

注 - 如果要更改特定接口的 IP 地址，请勿使用 `set-addressprop` 子命令，而是删除地址对象并使用新的 IP 地址创建新的地址对象。请参见“[删除或修改 IP 接口配置](#)” [66]。

例如，假设您要更改 IP 地址的网络掩码。该 IP 地址是在 IP 接口 `net3` 上配置的，由地址对象名称 `net3/v4` 标识。以下示例说明了如何修改网络掩码：

```
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/?        static    ok          127.0.0.1/8
net3/v4       static    ok          192.168.84.3/24

# 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
```

## 禁用、删除和修改 IP 接口配置

本节包含以下主题：

- [“删除 IP 接口配置”](#) [65]
- [“禁用 IP 接口配置”](#) [66]
- [“删除或修改 IP 接口配置”](#) [66]

## 删除 IP 接口配置

可以使用 `delete-ip` 子命令删除所配置的 IP 接口。在执行某些数据链路配置任务时，此命令尤为重要。例如，如果在某个数据链路上配置了 IP 接口，则重命名该数据链路将失败。在尝试重命名数据链路之前，需要使用 `ipadm delete-ip` 命令删除现有的 IP 配置。以下示例说明了用于执行此任务的命令：

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

有关其他信息，另请参见“[重命名数据链路](#)” [29]。要在重命名数据链路后重新配置 IP 接口，请参见[如何配置 IPv4 接口](#) [40]。

## 禁用 IP 接口配置

缺省情况下，使用 `ipadm create-ip` 命令创建接口时，IP 接口将激活并成为活动配置的一部分。在接口上创建第一个地址后，该接口将标记为 UP。

要从活动配置中删除接口而不销毁配置，请如下所示使用 `disable-if` 子命令。此子命令在内核中取消激活接口。

```
# ipadm disable-if -t interface
```

如下所示使 IP 接口可工作并使其标志显示为 UP：

```
# ipadm enable-if -t interface
```

---

提示 - 要显示 IP 接口的当前状态，请参见“[获取有关 IP 接口的信息](#)” [68]。

---

## 删除或修改 IP 接口配置

`ipadm delete-addr` 命令从 IP 接口删除特定的地址配置。在只需从系统中删除 IP 地址时或者在更改某个接口上配置的 IP 地址的过程中，此命令非常有用。如果要更改在某个接口上配置的 IP 地址，则必须先删除原始地址配置，然后再指定新地址配置。请参见[如何修改现有 IP 地址](#) [66]。

有关为接口创建 IP 地址的说明，请参见[如何配置 IPv4 接口](#) [40]。

---

注 - 一个接口可以有多个 IP 地址。每个地址由地址对象标识。要确保您删除的地址正确，必须知道地址对象。可以使用 `ipadm show-addr` 命令显示在某个接口上配置的 IP 地址。有关地址对象的说明，请参见[如何配置 IPv4 接口](#) [40]。有关显示 IP 地址的更多信息，请参见“[获取有关 IP 地址的信息](#)” [70]。

---

### ▼ 如何修改现有 IP 地址

以下过程介绍了重新配置系统的现有 IP 地址的步骤。

1. 成为管理员
2. 删除代表您要重新配置的 IP 地址的地址对象。

```
# ipadm delete-addr addrobj
```

3. 使用相同地址对象名称来指定新的 IP 地址。

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

要为系统添加其他接口，请参见《在 Oracle Solaris 11.2 中配置和管理网络组件》中的“如何配置 IPv4 接口”。

4. (可选) 如果需要，如下所示修改系统的主机名：

```
# hostname new-hostname
```

5. (可选) 如果子网掩码已更改，请修改子网条目。
6. (可选) 如果子网地址已更改，请更改缺省路由器的 IP 地址。  
有关说明，请参见“创建持久性（静态）路由” [54]。
7. 重新引导系统使更改生效。

## 监视 IP 接口和地址

可以使用 `ipadm` 命令监视并获取有关 IP 接口及其属性的信息。单独使用此命令可显示有关系统上 IP 接口的常规信息。但是，也可以使用各种子命令来限制要使用以下命令语法显示的信息：

```
ipadm show-* other-arguments interface
```

- 要仅获取接口信息，请使用 `show-if` 子命令。
- 要仅获取地址信息，请使用 `show-addr` 子命令。
- 要获取有关接口属性的信息，请使用 `show-ifprop` 子命令。
- 要获取有关地址属性的信息，请使用 `show-addrprop` 子命令。

有关 `ipadm show-*` 命令显示的所有字段的说明，请参见 [ipadm\(1M\)](#) 手册页。

## 获取有关 IP 接口的常规信息

可以通过 `ipadm` 命令全面了解系统的接口。仅使用该命令而不附带任何子命令可显示有关系统的所有 IP 接口的缺省信息。例如：

```
# 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   static  ok      --      10.132.146.233/23
net0/v4   dhcp    ok      --      10.132.146.234/23
ipmp0     ipmp    degraded --      --
ipmp0/v6  static  ok      --      2001:db8:1:2::4c08/128
net1      ip      failed  ipmp0   --
net1/v6   addrconf ok      --      fe80::124:4fff:fe58:1831/10
net2      ip      ok      ipmp0   --
net2/v6   addrconf ok      --      fe80::214:4fff:fe58:1832/10
iptun0    ip      ok      --      --
iptun0/v4 static  ok      --      172.16.111.5->172.16.223.75
iptun0/v6 static  ok      --      fe80::10:5->fe80::223:75
iptun0/v6a static  ok      --      2001:db8:1a0:7::10:5->2001:db8:7a82:64::223:75
    
```

以上输出显示了以下信息：

- IP 接口。
- 每个接口的种类。
- 每个接口的状态。
- 接口的状态：“独立”IP 接口，或底层接口（对于另一种接口配置类型）。在该示例中，net1 和 net2 是 ipmp0 的底层接口，如 UNDER 列所指示。
- 与接口关联的地址对象。地址对象标识特定 IP 地址。这些地址对象列在 NAME 标题下并进行了缩进，以便与接口名称进行区分。
- IP 地址的类型，已在 CLASS/TYPE 标题下进行了缩进，可以是 static、dhcp 等。
- 实际地址，在 ADDRESS 列下列出。

## 获取有关 IP 接口的信息

如需 IP 接口的信息，请使用 `ipadm show-if interface` 命令。如果您不指定接口，则会获取系统上所有接口的信息。

命令输出中的各个字段表示以下信息：

IFNAME	指显示其信息的接口。
CLASS	指接口的类，可以是四项之一： <ul style="list-style-type: none"> <li>■ ip 指 IP 接口</li> <li>■ ipmp 指 IPMP 接口</li> <li>■ vni 指虚拟接口</li> <li>■ loopback 指回送接口，它是自动创建的。除了回送接口，您可以手动创建其余 3 个接口类。</li> </ul>
STATE	指接口的状态，可以是以下状态之一：ok、offline、failed、down 或 disabled。

状态 `failed` (故障) 适用于 IPMP 组，可以指出现故障而无法承载通信流量的数据链路或 IP 接口。如果 IP 接口属于 IPMP 组，则 IPMP 接口可以继续使用组中其他活动 IP 接口接收和发送通信流量。

状态 `down` (关闭) 指由管理员切换为脱机模式的 IP 接口。

状态 `disable` (已禁用) 指通过使用 `ipadm disable-if` 命令取消激活的 IP 接口。

**ACTIVE** 指示接口是否用于承载通信，并且设置为 `yes` 或 `no`。

**OVER** 仅适用于 IPMP 接口类，指构成 IPMP 接口或组的底层接口。

下面是命令显示的信息的示例：

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

## 获取有关 IP 接口属性的信息

可以使用 `ipadm show-ifprop interface` 命令获取有关 IP 接口属性的信息。如果不指定属性或接口，则将显示有关系统上所有 IP 接口的全部属性的信息。

命令输出中各字段的含义如下：

<b>IFNAME</b>	指显示其信息的 IP 接口。
<b>PROPERTY</b>	指接口的属性。一个接口可以有几个属性。
<b>PROTO</b>	指属性所适用的协议，可以是 IPv4 或 IPv6。
<b>PERM</b>	指给定属性的允许的权限，可以是只读、只写或读写。
<b>CURRENT</b>	指活动配置中属性的当前值。
<b>PERSISTENT</b>	指系统重新引导时重新应用的属性值。
<b>DEFAULT</b>	指示指定属性的缺省值。
<b>POSSIBLE</b>	指可以分配给指定属性的值的列表。对于数字值，显示一个可接受值的范围。

注 - 如果任何字段值是未知的，例如，一个接口不支持正在请求其信息的属性，则值显示为问号 (?)。

以下示例说明了 `show-ifprop` 子命令所显示的信息类型：

```
# 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
```

## 获取有关 IP 地址的信息

如需 IP 地址的信息，请使用 `ipadm show-addr interface` 命令。如果不指定接口，则将显示有关系统上所有 IP 地址的信息。

命令输出中各字段的含义如下：

ADDROBJ	指定列出其 IP 地址的地址对象。
TYPE	指示 IP 地址为 <code>static</code> 、 <code>dhcp</code> 还是 <code>addrconf</code> 。 <code>addrconf</code> 值指示地址是使用无状态还是有状态地址配置获得的。
STATE	介绍了活动配置中地址对象的状态。有关这些值的完整列表，请参见 <a href="#">ipadm(1M)</a> 手册页。
ADDR	指定在接口上配置的 IP 地址。地址可以是 IPv4 或 IPv6 地址。隧道接口同时显示本地地址和远程地址。 有关隧道的更多信息，请参见《在 Oracle Solaris 11.2 中管理 TCP/IP 网络、IPMP 和 IP 隧道》中的第 5 章，“管理 IP 隧道”。

下面是 `show-addr` 子命令提供的信息的示例：

```
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
net0/v4       static    ok         192.168.84.3/24
tun0/v4       static    ok         172.16.134.1-->172.16.134.2
```

如果随命令指定了一个接口并且该接口具有多个地址，则将显示类似于以下内容的信息：

```
# ipadm show-addr net0
ADDROBJ      TYPE      STATE      ADDR
net0/v4       static    ok         192.168.84.3/24
net0/v4a      static    ok         10.0.1.1/24
net0/v4bc     static    ok         172.16.10.1
```

显示为 *interface/?* 的地址对象指示地址是由未使用 `libipadm` API 的应用程序在接口上配置的。此类应用程序不受 `ipadm` 命令控制，该命令要求地址对象名称使用 *interface/user-defined-string* 格式。有关指定 IP 地址的示例，请参见[如何配置 IPv4 接口 \[40\]](#)。

## 获取有关 IP 地址属性的信息

如需 IP 地址属性的信息，请使用 `ipadm show-addrprop addrobj` 命令。要列出所有属性，请省略 *addrobj* 选项。要列出所有 IP 地址的单个属性，请仅指定该属性。要显示特定地址的所有属性，请仅指定 *addrobj* 选项。

命令输出中各字段的含义如下：

ADDROBJ	指列出其属性的地址对象。
PROPERTY	指地址对象的属性。一个地址对象可以有几个属性。
PERM	指给定属性的允许的权限，可以是只读、只写或读写。
CURRENT	指当前配置中属性的实际值。
PERSISTENT	指系统重新引导时重新应用的属性值。
DEFAULT	指示指定属性的缺省值。
POSSIBLE	指可以分配给指定属性的值的列表。对于数字值，显示一个可接受值的范围。

下面是 `show-addrprop` 子命令显示的信息类型的示例：

```
# ipadm show-addrprop net1/v4
ADDROBJ  PROPERTY  PERM  CURRENT          PERSISTENT  DEFAULT          POSSIBLE
net1/v4  broadcast r-    192.168.84.255  --          192.168.84.255  --
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         --
```



## 在 Oracle Solaris 客户机上管理命名和目录服务

---

本章介绍了如何为 Oracle Solaris 主机客户机系统配置命名服务。有关命名和目录服务以及服务器端管理的全面概述，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》。

有关对命名和目录服务配置进行故障排除的信息，请参见《在 Oracle Solaris 11.2 中排除网络管理问题》中的第 3 章，“排除命名服务问题”。

本章包含以下主题：

- “命名服务配置中的新增功能” [73]
- “命名和目录服务配置概述” [74]
- “为本地文件模式配置系统” [76]
- “配置 DNS 客户机” [77]
- “配置 NIS 客户机” [80]
- “配置 LDAP 客户机” [81]

---

注 - 除非另有说明，否则本章中所述任务同时适用于 IPv4 和 IPv6 网络。

---

### 命名服务配置中的新增功能

下面是新增或更改的功能：

- 命名服务和系统配置已迁移到 SMF – 在此发行版中，命名服务通过服务管理工具 (Service Management Facility, SMF) 进行管理。先前修改特定文件来配置命名服务的行为不再有效，例如 `/etc/nsswitch.conf` 和 `/etc/resolv.conf` 不再起作用。此 Oracle Solaris 发行版中保留的传统配置文件仅供用来兼容以前的 Oracle Solaris 发行版。这些文件的内容是由与特定命名服务相关的 SMF 服务生成的。

如果不存在任何网络配置，则命名服务缺省为 `files only` 行为，而不是 `nis files`。应该始终启用 `svc:/system/name-service/cache` SMF 服务。另请注意，如果使用 SMF 命令更改了这些服务的配置，则必须启用并/或刷新服务以使更改生效。请参见 [svccfg\(1M\)](#) 和 [svcadm\(1M\)](#) 手册页。

- `resolv.conf` 错误检查功能 – 在命名服务迁移到 SMF 之前，会无提示地处理 `resolv.conf` 文件配置中的错误，并且不进行检测，也不产生任何警告。因此，`resolv.conf`

文件不根据其配置的方式运行。Oracle Solaris 11 中通过 SMF 模板引入了基本的错误检查功能，以正确报告错误状况。请参见 [resolv.conf\(4\)](#) 手册页。

- 域名系统 (Domain Name System, DNS) 服务器设置 – DNS 服务器的设置过程已更改。有关详细说明，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“管理 DNS (任务)”。

## 命名和目录服务配置概述

命名服务对存储的信息（例如主机名和地址、用户名、口令、访问权限等等）执行查找。提供该信息是为了让用户能够登录到其主机、对资源进行访问以及被授予权限。命名服务信息可以存储在文件、映射或各种形式的数据库文件中。这些系统信息库可以位于系统本地或位于基于中央网络的系统信息库或数据库中。如果没有中央命名服务，则每台主机都必须单独维护一份此信息的副本。如果集中存储所有数据，管理将变得更加轻松。命名服务对任何计算网络都是至关重要的。

支持以下命名和目录服务：

- Domain Name System, DNS (域名系统)

DNS 是在 TCP/IP 网络上实施的一个分层次的分布式数据库。它主要用来为 Internet 主机名查找 IP 地址以及为 IP 地址查找主机名。数据分布在网络中，并通过使用以句点分隔的名称（从右向左读取）进行定位。DNS 还用来存储其他与 Internet 相关的主机信息，例如邮件交换路由信息、位置数据以及可用的服务。该服务的分层特性使得既可以对本地域进行本地管理，同时又可以对连接到 Internet、内联网或这两者的其他域提供国际性覆盖能力。有关更多信息，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“DNS 命名服务的说明”。

DNS 协议的两个扩展由 `svc:network/dns/multicast` 服务进行管理。多播 DNS (Multicast DNS, mDNS) 在未安装传统 DNS 服务器的小型网络中实施 DNS。DNS 服务搜索 (DNS Service Discovery, DNS-SD) 对多播 DNS 进行了扩展，使之还提供简单的服务搜索（网络浏览）。请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“多播 DNS 和服务搜索的说明”。

- 网络信息系统 (Network Information System, NIS)

NIS（在本指南中发音为 "niss"）是独立于 DNS 开发的。NIS 的主要作用是通过对各种网络信息进行集中控制来更好地管理网络。NIS 存储有关网络、计算机名称和地址、用户、以及网络服务的信息。这种网络信息的集合被称为 NIS 名称空间。有关更多信息，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“NIS 命名服务的说明”。

- Lightweight Directory Access Protocol, LDAP (轻量目录访问协议)

LDAP 是用来访问目录服务器以使用分布式命名和其他目录服务的安全网络协议。该基于标准的协议支持一个分层次的数据库结构。在 UNIX 和多平台环境中都可以使用同一协议来提供命名服务。Oracle Solaris 支持 LDAP 与 Oracle Directory Server Enterprise Edition（之前称为 Sun Java System Directory Server）以及其他 LDAP 目录服务器结合使用。有关更多信息，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“LDAP 命名服务的说明”。

有关 Oracle Solaris 中的命名服务支持（服务器端和客户端）的全面概述，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的第 1 章，“关于命名和目录服务”和《使用 Oracle Solaris 11.2 目录和命名服务：LDAP》。

## 关于 name-service/switch SMF 服务

name-service/switch SMF 服务是一项可配置的选择服务，通过它可以指定为每种类型的网络信息使用的名称信息服务或源。

名称服务转换由调用以下任意接口的客户端应用程序使用：

- gethostbyname
- getpwuid
- getpwnam
- getaddrinfo

name-service/switch SMF 服务定义各个网络数据库应使用什么命名服务。此信息以前存储在 /etc/nsswitch.conf 文件中。虽然此文件仍然存在，但必须通过更改此 SMF 服务中的相应属性来修改此文件包含的配置设置。

可以按如下方式显示这些属性：

```
$ svccfg -s name-service/switch listprop config
config                                application
config/default                        astring    files
config/value_authorization            astring    solaris.smf.value.name-service.switch
config/password                       astring    "files ldap"
config/group                          astring    "files ldap"
config/host                           astring    "files dns"
config/automount                      astring    "files ldap"
```

config/default 属性指定要搜索哪些缺省源。如果特定数据库没有设置其自己的属性，则会使用缺省源。在前面的示例中，除了 password、group、host 和 automount 之外，所有数据库都使用本地文件作为源。如果还需要缺省源之外的某些源，则将为特定数据库创建一个属性。在本例中，将先在本地文件中搜索 password、groups 和 automount，然后在 LDAP 中搜索。先在本地文件中搜索主机查找，然后在 DNS 中搜索。

如果要更改您的系统上启用的命名服务，必须更新 name-service/switch SMF 服务的相应属性以使用正确的命名服务。例如，假定您的 name-service/switch 的配置与上例相似，然后您禁用了 LDAP 并改为启用 NIS。

- 在这种情况下，需要设置 name-service/switch 服务的以下属性才能使用文件和 NIS：
- config/password
- config/group
- config/automount

需要键入以下命令来正确设置这些属性：

```
# svccfg -s name-service/switch setprop config/password = astring: "files nis"
# svccfg -s name-service/switch setprop config/group = astring: "files nis"
# svccfg -s name-service/switch setprop config/automountconfig/password = astring:
  "files nis"
# svccfg -s name-service/switch:default refresh
```

有关完整详细信息，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的第 2 章，“关于名称服务转换”和《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“配置名称服务交换机”。

## 为本地文件模式配置系统

以本地文件模式运行时，系统会从本地目录中的文件获取所有 TCP/IP 配置信息。在网络客户机模式下，配置信息通过远程网络配置服务器提供给网络中的所有系统。

通常，网络上的下列服务器以本地文件模式运行：

- 网络配置服务器
- NFS 服务器
- 提供 NIS、LDAP 或 DNS 服务的名称服务器
- 邮件服务器
- 路由器

由于客户机能够以网络客户机模式或本地文件模式运行在任意指定网络上，因此您可以使用这些模式的组合，并且在这些模式中配置不同的系统。

### ▼ 如何以本地文件模式配置系统

1. 成为管理员。
2. 使用指定的 IP 地址配置系统的 IP 接口。  
请参见《在 Oracle Solaris 11.2 中配置和管理网络组件》中的“如何配置 IPv4 接口”。

3. 验证是否正确设置了主机名。

```
# hostname
```

有关更多信息，请参见 `hostname(1)` 手册页。

4. 验证 `/etc/inet/hosts` 中的条目是否为最新的。  
Oracle Solaris 会为主网络接口、回送地址和在安装期间配置的任何其他接口（如果适用）创建条目。如果条目不是最新的，则会为安装后添加到系统的任何网络接口添加 IP 地址和对应名称。

5. 将系统的全限定域指定为 `nis/domain` SMF 服务的一个属性。

例如，可以将 `deserts.worldwide.com` 指定为 `nis/domain` SMF 服务的 `domainname` 属性的值，如下所示：

```
# domainname domainname
```

此步骤使更改成为永久性的。

6. 添加路由信息。

---

注 - 如果您使用 DHCP 服务，则可以跳过此步骤。

---

有关说明，请参见“配置路由” [52]。

7. 添加网络掩码信息（如果适用）。

---

注 - 如果您使用 DHCP 服务，则可以跳过此步骤。

---

- a. 在 `/etc/inet/netmasks` 文件中键入网络号和网络掩码。

要创建项，请使用格式 `network-number netmask`。例如，要指定 C 类网络号 `192.168.83`，应键入以下信息：

```
192.168.83.0    255.255.255.0
```

对于 CIDR 地址，请将网络前缀转换为等效的点分十进制表示法格式。例如，键入以下信息可以表示 CIDR 网络前缀 `192.168.3.0/22`。

```
192.168.3.0    255.255.252.0
```

- b. 在 `name-service/switch` 属性中更改网络掩码的查找源，以便只搜索本地文件，然后刷新实例。

```
# svccfg -s name-service/switch setprop config/netmask = astring: "files"
# svccfg -s name-service/switch:default refresh
```

8. 重新引导系统。

## 配置 DNS 客户机

DNS 有两个部分：一个用来提供答复的服务和一个用来查询服务的客户机。在 Oracle Solaris 中，缺省 DNS 服务是由互联网系统协会 (Internet Systems Consortium, ISC) 的伯克利互联网名称域 (Berkeley Internet Name Domain, BIND) 及其关联的 `named` 服务器守护进程提供的。DNS 客户机包含很多实用程序和库。

有关与任务相关的其他信息，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“管理 DNS（任务）”。

## ▼ 如何启用 DNS 客户机

1. 成为管理员。
2. 列出要搜索的域和 DNS 名称服务器的 IP 地址，然后更新 SMF 系统信息库。例如：

```
# svccfg -s network/dns/client
svc:/network/dns/client> setprop config/search = astring: ("example.com" "sales.
example.com")
svc:/network/dns/client> setprop config/nameserver = net_address: (192.168.1.10
192.168.1.11)
svc:/network/dns/client> select network/dns/client:default
svc:/network/dns/client:default> refresh
svc:/network/dns/client:default> quit
```

确保刷新服务以使更改生效。

3. 更新名称服务转换信息以使用 DNS。  
第一个命令更新 SMF 系统信息库中 DNS 配置信息。

```
# svccfg -s system/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> quit
```

4. 启动运行 DNS 客户机所需的服務。
5. 使用以下一个或所有命令来验证 DNS 客户机是否已启用：

```
# dig knownserver.example.com
# getent hosts knownserver.example.com
```

dig 命令在单独使用时将检查 DNS 客户机是否已启用。getent hosts 命令验证 /etc/nsswitch.conf 文件是否使用了 DNS 客户机。

### 例 12 同时为客户机设置多个 DNS 选项

以下示例说明了如何设置多个 /etc/resolv.conf 选项。

```
# svccg
```

```

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

```

## 启用多播 DNS

要使多播 DNS (Multicast DNS, mDNS) 和 DNS 服务搜索正常运行，必须在将要参与 mDNS 的所有系统上部署 mDNS。mDNS 服务用来通告系统上提供的服务的可用性。

在启用 mDNS 之前，请确保系统上安装了软件包。如果需要，请按照以下方法安装软件包：

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

在启用 mDNS 的过程中，将首先更新名称服务转换信息。为了能够解析本地主机，必须如下所示更改 name-service/switch SMF 服务的 config/host 属性，将 mdns 包括为源：

```

# /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

```

如下所示启用 mDNS SMF 服务：

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

使用此方法启用 mDNS 时，您的更改在升级和重新引导后将持久存在。有关更多信息，请参见 [svcadm\(1M\)](#) 手册页。

## 为 DNS 通告资源

可以使用 `dns-sd` 命令浏览和搜索服务，这与使用 `ping` 或 `traceroute` 命令的方式相似。`dns-sd` 命令主要以交互方式使用，这主要是因为它的命令行参数和输出格式在以后可能会更改，这使得通过 shell 脚本调用该命令具有不可预测性和风险。此外，DNS 服务搜索 (DNS service discovery, DNS-SD) 的异步性质使其无法轻易地用于面向脚本的编程。

有关示例，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“为 DNS 通告资源”。另请参见 [dns-sd\(1M\)](#) 手册页。

## 配置 NIS 客户机

NIS 是一种分布式命名服务，用于标识和定位网络对象及资源。NIS 通过某种传输协议以独立于介质的方式为网络范围内的信息提供了统一的存储和检索方法。NIS 是独立于 DNS 开发的，并且其侧重点也稍有不同。DNS 侧重于使用计算机名称而不是数字 IP 地址来简化通信，而 NIS 侧重于对各种网络信息进行集中控制来更好地管理网络。

通过运行 NIS，您可以将管理数据库（映射）分布在多个服务器（主服务器和从属服务器）之间。您可以通过一种自动而且可靠的方式从一个集中位置更新这些数据库，以确保整个网络中的所有客户机都一致地共享相同的命名服务信息。有关详细概述，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的第 5 章，“关于网络信息服务”。

有两种 SMF 服务在系统上提供 NIS 客户机服务。可以使用 `svcadm` 命令启用、禁用、重新启动和刷新这些服务。显示系统上 NIS 服务的状态，如下所示：

```
# svcs \*nis\*
STATE          STIME      FMRI
online         Oct_09    svc:/network/nis/domain:default
online         Oct_09    svc:/network/nis/client:default
```

有关与任务相关的其他信息，请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的第 6 章，“设置和配置网络信息服务”。

### ▼ 如何在广播模式下配置 NIS 客户机

广播模式是建立 NIS 客户机的最简单方法。当您启动 `nis/client` SMF 服务时，此服务会运行 `ybind` 命令，该命令在本地子网中搜索 NIS 服务器。如果找到了一个子网，则 `ybind` 命令将绑定到该子网。这种搜索被称为广播。如果在客户机的本地子网中不存在 NIS 服务器，则 `ybind` 将无法绑定，并且客户机无法从 NIS 服务获取名称空间数据。请参见[如何使用特定的 NIS 服务器配置 NIS 客户机 \[81\]](#)。

1. 成为管理员。
2. 设置 NIS 域名。  

```
# domainname example.com
```
3. 如果需要，对名称服务转换进行更改。  
 请参见《使用 Oracle Solaris 11.2 目录和命名服务：DNS 和 NIS》中的“配置名称服务交换机”。
4. 启动 NIS 客户机服务。

```
# svcadm enable network/nis/domain
# svcadm enable network/nis/client
```

## ▼ 如何使用特定的 NIS 服务器配置 NIS 客户机

开始之前 以下过程要求在步骤 3 中指定的主机名（服务器）可以由 DNS 进行解析。如果您未使用 DNS，并且键入了主机名而不是 IP 地址，请确保在客户机上的 `/etc/hosts` 文件中为每个 NIS 服务器添加相应条目。有关更多信息，请参见 [ypinit\(1M\)](#) 手册页。

1. 成为管理员。

2. 设置 NIS 域。

```
# domainname example.com
# svcadm enable network/nis/domain
```

3. 运行客户机配置脚本。

```
# ypinit -c
```

系统将提示您列举客户机要从中获取命名服务信息的 NIS 服务器。您可以根据需要列出主服务器和任意多的从属服务器。列出的服务器可以位于域中的任意位置。最好先列出离计算机最近的服务器（从网络角度来说），然后再列出网络中处于更远距离的服务器。

4. 启用 NIS 客户机。

```
# svcadm enable network/nis/client
```

## ▼ 如何禁用 NIS 客户机服务

1. 成为管理员。

2. 如下所示停止 NIS 客户机服务：

```
# svcadm disable network/nis/domain
# svcadm disable network/nis/client
```

## 配置 LDAP 客户机

要让 Oracle Solaris 客户机使用 LDAP 作为命名服务，必须满足以下要求：

- 客户机的域名必须由 LDAP 服务器提供服务。
- 对于必需的服务，名称服务转换必须指向 LDAP。
- 必须为客户机配置定义其行为的所有给定参数。
- `ldap_cachemgr` 必须正在客户机上运行。

- 至少有一台要为其配置客户机的服务器已启动并正在运行。

`ldapclient` 命令是设置 LDAP 客户机的关键，因为该命令执行除启动服务器之外的上述所有步骤。

为网络配置使用固定模式时，在客户机系统上设置 LDAP 的最简单方式是启用 `DefaultFixed` 配置文件并执行持久性网络配置。然后，可以使用 `ldapclient` 命令，通过使用配置文件或手动设置来完成 LDAP 设置。有关更多信息，请参见 [ldapclient\(1M\)](#) 手册页。

有关 LDAP 的详细概述，请参见《[使用 Oracle Solaris 11.2 目录和命名服务：LDAP](#)》。

## 导入命名服务配置

`nscfg` 命令将 `name-service` 切换组件的传统文件配置转移到 SMF 系统信息库。升级到 Oracle Solaris 11 时，系统的命名服务配置将自动迁移到 SMF。但是，如果需要，您可以使用 `nscfg` 命令手动将此配置迁移到 SMF。

以下命令导入传统文件，然后转换配置并将其推送到 SMF：

```
# /usr/sbin/nscfg import -f FMR/
```

要利用已有的 `resolv.conf` 文件的信息来填充 DNS 配置，使用 `nscfg` 命令是最简单的方式。在以下示例中，`nscfg` 命令读取 `/etc/resolv.conf` 文件中的信息，转换这些信息，然后将信息存储在 `svc:/network/dns/client` SMF 服务中：

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

更改系统的命名服务时，您还需要相应地修改名称服务转换信息，并可能需要更新名称服务高速缓存中任何过时的信息，如以下示例中所示：

```
# 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
```

有关更多信息，请参见 [nscfg\(1M\)](#) 手册页。

## 重置 SMF 命名服务配置

您可以按照以下方法将 SMF 命名服务的配置属性重置回 `files only` 模式：

```
# /usr/sbin/nscfg unconfig FMRI  
# svcadm refresh name-service/switch
```

例如，您可以恢复在“[导入命名服务配置](#)” [82]中对 SMF 配置所做的更改，如下所示：

```
# 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
```



## 关于在 Oracle Solaris 中管理基于配置文件的网络配置

---

本章概述了基于配置文件的网络配置，这种配置主要在使用反应性模式时使用。反应性模式支持用于管理 Oracle Solaris 系统上网络配置的各种类型的配置文件。此模式主要用于笔记本个人电脑以及网络条件可能经常发生变化的情况。在反应性模式中，网络守护进程 (nwmnd) 监视系统的状态并在条件发生变化时动态调整网络配置。

有关使用固定模式（公司环境中最常用的网络管理模型）的信息，请参见第 2 章在 [Oracle Solaris 中管理数据链路配置](#) 和第 3 章在 [Oracle Solaris 中配置和管理 IP 接口和地址](#)。

有关配置和管理网络配置文件的逐步说明，请参见第 6 章在 [Oracle Solaris 中管理基于配置文件的网络配置](#)。

本章包含以下主题：

- “关于反应性模式” [85]
- “关于基于配置文件的网络配置” [86]
- “使用基于配置文件的网络配置的准则” [90]
- “使用基于配置文件的网络配置的安全要求” [92]
- “基于配置文件的网络配置如何与其他 Oracle Solaris 功能一起使用” [93]

### 关于反应性模式

反应性模式意味着系统自动适应网络条件的任何变化，然后调整系统的当前网络配置，而无需任何手动重新配置。例如，如果有线网络接口状态变为拔出、有新的无线网络可用或者您更改了物理位置，则系统会相应地进行调整其网络配置。

Oracle Solaris 中的反应性网络配置策略的主要特点是灵活，它使得系统的网络配置可以动态更改，以响应不同的网络事件或您的请求。此类型的网络配置最适合笔记本个人电脑使用以及网络条件经常发生变化的情况。使用反应性模式时，系统的基本以太网和 WiFi 配置将自动执行。系统在启动时自动连接到有线或无线网络，并在桌面中显示有关

当前活动网络连接状态的通知。可以使用属性来配置反应性配置文件，这些属性确定在什么条件下启用哪个特定配置文件。这些属性使网络管理守护进程 `nwamd` 能够根据需要对系统动态地应用配置文件的配置。

反应性网络配置变化通常由以下事件和活动触发：

- 连接或断开以太网电缆
- 连接或断开无线局域网 (wireless local area network, WLAN) 网卡
- 引导系统时，有线接口或无线接口可用
- 从挂起状态恢复时，有线接口或无线接口可用（如果支持）
- 获得或失去 DHCP 租用

在反应性模式下，使用两个命令管理网络配置：用于对配置文件进行网络配置更改的 `netcfg` 命令，以及用于显示有关配置文件的信息及启用和禁用配置文件的 `netadm` 命令。有关完整说明，请参见 [netcfg\(1M\)](#) 和 [netadm\(1M\)](#) 手册页。有关与任务相关的信息，请参见第 6 章在 [Oracle Solaris 中管理基于配置文件的网络配置](#)。

另一方面，固定模式与反应性模式相反。使用固定模式时，网络守护进程在系统中实例化一个特定的网络配置，但不会根据网络条件的变化来自动调整该配置。有关固定模式的更多信息，请参见[“关于网络配置模式” \[14\]](#)。

有关支持的所有网络管理命令以及何时使用的更多信息，请参见[“Oracle Solaris 网络管理命令” \[15\]](#)。

## 关于基于配置文件的网络配置

Oracle Solaris 提供了一组预先确定的系统定义配置文件，并能够使用您指定的属性和激活条件创建各种类型的用户定义反应性配置文件，以满足您的特定网络需求。用户定义的配置文件可用于简化系统上数据链路和 IP 地址的基本配置，以及定义更复杂的系统级网络配置，例如，命名服务、IP 过滤器和 IP 安全 (IP Security, IPsec) 配置。

支持下列配置文件类型：

- **网络配置文件 (network configuration profile, NCP)** – NCP 是主要配置文件类型，可用于指定网络数据链路和 IP 接口的配置。NCP 配置有属性值，这些属性值指定在系统上激活了特定 NCP 时如何配置网络。NCP 可以是反应性的，也可以是固定的。您可以配置多个反应性 NCP，但 Oracle Solaris 只支持一个名为 `DefaultFixed` 的固定 NCP。
- **网络配置单元 (Network Configuration Unit, NCU)** – 在 NCU 中指定定义 NCP 的单个配置信息（属性）。一个 NCU 表示一个物理链路或一个接口，且包含指定该链路或接口的配置属性。
- **位置配置文件** – 位置配置文件（也称为“位置”）指定系统级的网络配置，例如，命名服务、域、IP 过滤器配置和 IPsec 配置。
- **外部网络修饰器 (External Network Modifier, ENM)** – ENM 是管理应用程序的配置文件，这些应用程序负责创建系统主网络配置外部的网络配置，例如 VPN 应用程序。

- 已知 WLAN – 已知 WLAN 是存储系统所搜索到的无线网络的相关信息的配置文件。

## 配置文件类型说明

下面详细说明了 Oracle Solaris 发行版中支持的各种配置文件类型。

### NCP 说明

NCP 定义系统特定的网络配置，例如数据链路及 IP 接口和地址。组成各 NCP 的不同网络配置单元 (network configuration unit, NCU) 指定如何配置不同的网络链路和接口，例如，应调用哪个或哪些接口？应在什么条件下调用接口？以及如何获取接口的 IP 地址？

Automatic NCP 表示系统中当前的所有网络链路和接口。如果添加或移除网络设备，Automatic NCP 的内容将发生变化。Automatic NCP 提供对利用 DHCP 地址自动配置的配置文件的访问，这使其可以从系统获取 IP 地址。此 NCP 还实现如下链路选择策略：有线链路优先，无线链路次之。如果需要指定备用 IP 配置策略或备用链路选择策略，您需要在系统中创建另一个 NCP。无法删除 Automatic NCP。可以复制此 NCP 并对副本进行更改。请参见例 19 “通过克隆 Automatic NCP 来创建 NCP”。

### NCU 说明

NCU 包含定义 NCP 的属性值。NCU 表示系统中的单个物理链路和接口。配置用户定义的 NCP 的过程包括创建那些指定应如何配置每个链路和接口以及在什么条件下进行配置的 NCU。

有两种类型的 NCU：

- 链路 NCU – 表示物理设备（开放系统互连 (Open Systems Interconnection, OSI) 模型中的第 2 层实体)
- 接口 NCU – 表示 IP 接口（第 3 层实体)

链路 NCU 表示以下数据链路层类：

- 聚合
- 桥接程序
- Etherstub
- 基于 IB 的以太网 (Ethernet over IB, EoIB)
- 物理链路（以太网或 WiFi)
- 隧道
- 虚拟可扩展局域网 (Virtual eXtensible Local Area Network, VXLAN)
- 虚拟局域网 (Virtual Local Area Network, VLAN)
- 虚拟网络接口卡 (vNIC)

接口 NCU 表示以下 IP 层类：

- IP 接口
- IPMP 接口
- VNI 接口

有关可以为各个对象类型设置的属性的信息，请参见 [netcfg\(1M\)](#) 手册页。

## 位置配置文件说明

位置配置文件（也可简称为位置）包括网络配置信息，例如，在该位置处于活动状态时一起应用以指定系统级网络配置的命名服务和防火墙设置。由于位置并不一定与物理位置对应，您可以设置多个位置配置文件以满足不同网络需求。例如，可以在连接到公司内联网时使用一个位置，在通过位于办公室中的无线访问点连接到公共 Internet 时使用另一个位置。

缺省情况下，系统预定义了三个位置：

- **DefaultFixed**

只要 DefaultFixed NCP 处于活动状态，就会启用 DefaultFixed 位置。无法使用 netcfg 命令直接修改 DefaultFixed 位置。作为启用 DefaultFixed NCP 的过程的一部分启用此位置时，相关服务管理工具 (service management facility, SMF) 属性将更新以反映位置的设置。当系统关闭或者启用另一个位置时，相关 SMF 属性将作为 DefaultFixed 位置配置的一部分保存。

- **自动**

如果有网络可用但没有其他位置取代它，则激活 Automatic 位置。可以使用 netcfg 命令修改 Automatic 位置。

---

注 - Automatic 位置不应与 Automatic NCP 混为一谈。Automatic 位置在系统进行初始的网络配置后定义系统级网络属性。Automatic NCP 指定系统中的链路和接口网络配置。

---

- **NoNet**

NoNet 位置具有特定的激活条件。当没有一个本地接口具有指定的 IP 地址时，系统将此位置应用到独立系统。可以使用 netcfg 命令修改 NoNet 位置。

用户定义的位置和系统定义的位置一样，只是用户定义的位置配置有您指定的定制值，而系统定义的位置具有预设值。

## ENM 说明

通过 ENM 可以指定应用程序或脚本应何时执行网络配置，对于在 NCP 和位置配置文件中指定的配置来说，这是外部网络配置。ENM 也可以定义为在启用或禁用 ENM 时直接

修改网络配置的服务或应用程序。可以指定启用或禁用 ENM 的条件。还可以手动启用或禁用 ENM。对于 NCP 或位置配置文件，在任何给定时间系统中只能有一个配置文件类型处于活动状态。与 NCP 或位置配置文件不同，多个 ENM 可以在同一时间在系统中处于活动状态。在任何给定时间在系统中处于活动状态的 ENM 不一定依赖于同时在系统中处于启用状态的 NCP 或位置配置文件。

不过，有一些外部应用程序和服务，您可以为其创建 ENM，明显的一个例子就是 VPN 应用程序。在系统中安装并配置 VPN 后，您可以创建一个 ENM，使其在您指定的条件下自动激活和取消激活 VPN。

---

注 - 反应性网络配置模式无法自动检测能够直接修改系统上网络配置的外部应用程序。要管理 VPN 应用程序（或任何外部应用程序或服务）的激活或取消激活，必须先安装该应用程序，然后通过命令行界面 (command-line interface, CLI) 或网络管理 GUI 为其创建 ENM。

---

存储或跟踪有关由 ENM 执行的任何网络配置的持久性信息时，采用的方式与存储关于 NCP 或位置配置文件的信息的方式不完全相同。然而，系统能够记录到外部发起的网络配置，然后根据由 ENM 对系统所做的任何配置更改，重估应激活哪个位置配置文件，并随后激活该位置。例如，当特定 IP 地址正在使用时，切换到一个根据条件激活的位置。如果 `svc:/network/physical:default` 服务在任何时间重新启动，将恢复由活动 NCP 指定的网络配置。ENM 也会重新启动，在该过程中，可能会破坏网络配置并重新创建。

## 已知 WLAN 的说明

已知 WLAN 是用于管理系统已知的无线网络的配置文件。然后，由系统维护这些已知无线网络的全局列表。此信息用于确定尝试连接到可用无线网络时采取的顺序。如果已知 WLAN 列表中有一个无线网络可用，则系统会自动连接到该网络。如果有两个或更多个已知无线网络可用，则系统将尝试连接到优先级最高（数字最小）的无线网络。您连接到的任何新的无线网络会自动添加到已知 WLAN 列表的顶部，成为当前优先级最高的无线网络。

缺省行为是使最近所连接到的 WLAN 优先于以前连接到的那些 WLAN。无论什么时候，任何已知 WLAN 都不能具有相同的优先级。如果将新 WLAN 添加到列表中时使用与某个现有 WLAN 相同的优先级值，该现有条目将移到较低的优先级值。随后，列表中每一个其他 WLAN 的优先级值动态移到较低的优先级值。

还可以将一个密钥名称与已知 WLAN 相关联。通过密钥名称可以使用 `dladm create-secobj` 命令创建您自己的密钥。然后，您可以通过将安全对象名称添加到已知 WLAN `keyname` 属性中，将此密钥与 WLAN 相关联。有关更多信息，请参见 [dladm\(1M\)](#) 手册页。

有关从命令行管理 WLAN 的更多信息，请参见[“在反应性模式下管理已知 WLAN” \[130\]](#)。

## 系统定义的配置文件和用户定义的配置文件

Automatic NCP 是系统定义的配置文件，由与系统中存在的每个物理链路相对应的一个链路 NCU 和一个接口 NCU 组成。此 NCP 中的 NCU 激活策略是先选择已连接的有线链路，再选择无线链路，并对每个已启用的链路同时激活 IPv4 和 IPv6。DHCP 用于获取 IPv4 地址。无状态自动配置和 DHCP 用于获取 IPv6 地址。插入新链路或从系统中删除链路时，Automatic NCP 将动态更改。与插入或删除的链路对应的所有 NCU 也将同时添加或删除。由 nwamd 守护进程自动更新该配置文件。

处于活动状态时，Automatic NCP 实现以下基本策略：

- 通过使用 DHCP 来配置所有可用（已连接的）以太网接口。
- 如果没有连接以太网接口，或者如果以太网接口无法获取 IP 地址，则启用一个无线接口，自动连接到已知 WLAN 列表中最合适的 WLAN。请参见“[已知 WLAN 的说明](#)” [89]。
- 在获得至少一个 IPv4 地址之前，保持 NoNet 位置处于活动状态。请参见“[位置配置文件说明](#)” [88]。此位置提供了一组严格的 IP 过滤器规则，这些规则限制仅传递与 IP 地址获取相关的数据（DHCP 和 IPv6 autoconf 消息）。NoNet 位置的所有属性（除激活条件之外）都可以修改。
- 至少有一个 IP 地址指定给其中一个系统接口时，将激活 Automatic 位置。此位置没有 IP 过滤器或 IPsec 规则。此位置应用从 DHCP 服务器获取的域名系统 (domain name system, DNS) 配置数据。和 NoNet 位置一样，Automatic 位置的所有属性（除激活条件之外）都可以修改。

可以有选择地配置用户定义的 NCP。必须在指定的 NCP 中显式添加和删除 NCU。您还可以创建不与系统中当前的任何链路相关联的 NCU。此外，可以为用户定义的 NCP 确定策略。例如，您可以允许在给定的时间在系统中启用多个链路和接口，还可以指定 NCU 和静态 IP 地址之间的不同的依赖关系。

## 使用基于配置文件的网络配置的准则

基于配置文件的网络配置遵循以下准则：

- 在任意时刻，系统上只能有一个网络配置文件 (network configuration profile, NCP) 和一个位置配置文件处于活动状态。系统中所有其他现有 NCP 均不工作。
- 活动的 NCP 既可以是反应性的，也可以是固定的 (DefaultFixed)。系统使用反应性 NCP 监视网络配置以适应系统网络环境中的变化。使用 DefaultFixed NCP（系统唯一的固定配置文件）时，网络配置将会实例化，但不受监视。
- NCP 的不同属性的值构成一个策略，控制配置文件如何管理系统的网络配置。
- 对 NCP 属性的任何更改会立即作为新的属性值实现，只要该配置文件处于活动状态，就会成为用于管理网络配置的配置文件的策略的一部分。
- 如果系统正在使用反应性模式，则管理网络配置的活动 NCP 为 Automatic NCP 或您所创建的用户定义反应性 NCP。当反应性 NCP 处于活动状态时，可使用 netcfg 和 netadm 命令管理网络配置。

如果系统正在使用固定模式，则管理其网络配置的活动 NCP 始终为 DefaultFixed。当此 NCP 处于活动状态时，可使用 `dladm` 和 `ipadm` 命令管理网络配置。有关更多信息，请参见第 2 章在 [Oracle Solaris 中管理数据链路配置](#) 和第 3 章在 [Oracle Solaris 中配置和管理 IP 接口和地址](#)。

## 配置文件激活策略

配置文件的激活模式可以为手动、自动或有条件。当反应性配置文件处于活动状态时，网络环境中发生的更改将导致系统重估网络配置，然后做出“最佳推测”，根据当前条件决定要激活的反应性 NCP 和位置。网络条件的更改包括连接或断开以太网电缆、获得或失去 DHCP 租用以及检测到新的无线网络。系统中必须始终有一个 NCP 和一个位置处于活动状态。

反应性网络配置模式使您可以指定反应性 NCP 的激活策略。此策略确定何时启用 NCU。每个位置配置文件中还包含定义激活条件的属性。

NCU、位置和 ENM 均具有 `activation-mode` 属性。这些配置文件类型中的每一个所允许使用的值有所不同。此外，针对每种配置文件类型验证 `activation-mode` 属性的方式不同，每个配置文件类型的启用条件也不同。

---

注 - NCU 的 `activation-mode` 属性可以设置为 `manual` 或 `prioritized`。位置配置文件的 `activation-mode` 属性可以设置为 `manual`、`conditional-any`、`conditional-all` 或 `system`。

---

NCP 激活策略通过为每个 NCU 指定的属性和条件来执行。例如，您可以指定如下策略：“有线连接优先于无线连接”或“一次激活一个接口”。NCP 如何启用以及何时启用是在为每个 NCU 类型设置的属性中定义的。有关激活条件的更多信息，请参见 [netcfg\(1M\)](#) 手册页。

---

注 - 接口 NCU 与底层链路 NCU 相关联。每个接口 NCU 在与其相关联的链路 NCU 被启用时转为活动状态。对底层链路 NCU 的依赖性永远不会删除。如果启用了接口 NCU 而未启用与其相关联的链路 NCU，则在启用该接口的底层链路 NCU 之前，该接口 NCU 不会实际转变为活动状态。

---

## 配置文件激活模式

用户定义的 NCU、位置配置文件和 ENM 都具有 `activation-mode` 属性。`activation-mode` 属性在您使用 `netcfg` 命令创建或修改配置文件时设置。NCP 不具有 `activation-mode` 属性。所有 NCP 都为手动启用。

下表说明了不同配置文件类型的 `activation-mode` 属性的可能值。

表 1 activation-mode 属性值

配置文件类型	activation-mode 值
NCU	manual 或 prioritized
位置	manual、conditional-any、conditional-all 或 system
ENM	manual、conditional-any 或 conditional-all

有关启用和禁用配置文件的更多信息，请参见[“启用和禁用配置文件” \[95\]](#)。

## 使用基于配置文件的网络配置的安全要求

netcfgd 守护进程控制存储所有网络配置信息的系统信息库。netcfg 命令、网络管理 GUI 和 nwamd 守护进程都会向 netcfgd 守护进程发送访问系统信息库的请求。

当前网络配置实现使用以下授权执行特定的任务：

- solaris.network.autoconf.read – 允许读取网络配置文件数据，由 netcfgd 守护进程验证。
- solaris.network.autoconf.write – 允许写入网络配置文件数据，由 netcfgd 守护进程验证。
- solaris.network.autoconf.select – 允许应用新的配置数据，由 nwamd 守护进程验证。
- solaris.network.autoconf.wlan – 允许写入已知 WLAN 配置数据。

这些授权在 auth\_attr 数据库中注册。请参见 [auth\\_attr\(4\)](#) 手册页。

solaris.network.autoconf.read 授权包含在 "Basic Solaris User" (基本 Solaris 用户) 权限配置文件中，该配置文件缺省指定给所有用户。具有此授权的任何用户因此能够查看当前网络状态和所有网络配置文件的内容。

提供了两个其他权限配置文件："Network Autoconf User" (网络自动配置用户) 和 "Network Autoconf Admin" (网络自动配置管理员)。  
 "Network Autoconf User" (网络自动配置用户) 配置文件具有 read、select 和 wlan 授权。  
 "Network Autoconf Admin" (网络自动配置管理员) 配置文件添加了 write 授权。  
 "Network Autoconf User" (网络自动配置用户) 配置文件将指定给 "Console User" (控制台用户) 配置文件。缺省情况下，登录到控制台的任何人都可以查看、启用和禁用配置文件。因为没有向 "Console User" (控制台用户) 配置文件指定 solaris.network.autoconf.write 授权，所以拥有该授权的用户不能创建或修改 NCP、NCU、位置或 ENM。不过，"Console User" (控制台用户) 配置文件可以查看、创建和修改 WLAN。

具有 "Basic Solaris User" (基本 Solaris 用户) 权限配置文件的任何用户都可以使用 netcfg 和 netadm 命令查看网络配置文件。缺省情况下该配置文件会指定给所有用户。

具有 "Network Autoconf User" (网络自动配置用户) 或 "Console User" (控制台用户) 配置文件的任何用户还可以使用 `netadm` 命令启用配置文件。"Console User" (控制台用户) 配置文件会自动指定给从 `/dev/console` 登录到系统的用户。

要使用 `netcfg` 命令修改网络配置文件, 需要 `solaris.network.autoconf.write` 授权或 "Network Autoconf Admin" (网络自动配置管理员) 配置文件。

例如, 如下所示确定与 "Console User" (控制台用户) 权限配置文件相关联的特权:

```
$ profiles -p "Console User" info
name=Console User
desc=Manage System as the Console User
auths=solaris.system.shutdown,solaris.device.cdrw,solaris.devinde.mount.removable,
solaris.smf.manage.vbiosd,solaris.smf.value.vbiosd
profiles=Suspend To RAM,Suspend To Disk,Brightness,CPU
Power Management,Network Autoconf User,Desktop Removable Media User
help=RtConsUser.html
```

## 基于配置文件的网络配置如何与其他 Oracle Solaris 功能一起使用

反应性网络配置模式可与其他 Oracle Solaris 网络技术配合使用, 如下所示:

- 虚拟机: Oracle VM Server for SPARC (先前的 Logical Domains) 和 Oracle VM VirtualBox
 

反应性配置文件可用于 Oracle Solaris 主机操作系统和客操作系统。但是, 反应性网络配置模式只管理属于指定虚拟机的接口, 不会干扰系统上的其他虚拟机。
- Oracle Solaris 区域和栈实例
 

反应性配置文件可以在全局区域中使用, 或者在专用栈非全局区域中使用。但是, 您不能为共享栈区域配置反应性配置文件, 因为共享栈区域的网络配置始终在全局区域中进行管理。
- 动态重新配置 (Dynamic Reconfiguration, DR)
 

在具有动态重新配置 (dynamic reconfiguration, DR) 功能和热插拔功能的系统上, 系统的网络配置才支持这些功能。可以使用这些功能添加或移除设备, 而不论当前哪个 NCP 处于活动状态 (反应性 NCP 还是 DefaultFixed NCP)。但是, 系统的行为因当前活动的 NCP 类型而异。

Automatic NCP 或其他反应性 NCP 处于活动状态并且有设备插入时, NCP 会自动为新添加的设备创建 IP 配置。如果在反应性配置文件当前处于活动状态时从系统中移除了设备, 则该设备的 IP 接口将取消配置。DefaultFixed NCP 在系统上处于活动状态时, 必须在添加设备之后显式配置 IP 接口。还必须在移除设备之前显式删除 IP 配置。

有关动态配置设备的更多信息, 请参见《在 Oracle Solaris 11.2 中管理设备》。有关在使用固定配置文件时执行动态重新配置的更多信息, 请参见《在 Oracle Solaris 11.2 中配置和管理网络组件》中的“如何使用动态重新配置更换网络接口卡”。



## 在 Oracle Solaris 中管理基于配置文件的网络配置

---

本章介绍了如何使用各种类型的配置文件来管理网络配置。有关基于配置文件的网络配置和反应性模式的概述，请参见[第 5 章 关于在 Oracle Solaris 中管理基于配置文件的网络配置](#)。

有关在使用固定模式时配置数据链路和 IP 接口的信息，请参见[第 2 章 在 Oracle Solaris 中管理数据链路配置](#)和[第 3 章 在 Oracle Solaris 中配置和管理 IP 接口和地址](#)。

本章包含以下主题：

- “启用和禁用配置文件” [95]
- “配置配置文件” [97]
- “管理配置文件” [108]
- “从桌面管理网络配置” [120]

### 启用和禁用配置文件

可以使用 `netadm` 命令启用和禁用所有配置文件，不论配置文件的类型如何，也不论配置文件是固定的还是反应性的。基本命令语法如下所示：

```
# netadm enable [ -p profile-type ] [ -c ncu-class ] profile-name
```

例如，如下所示启用系统定义的自动 NCP：

```
# netadm enable -p ncp Automatic
```

有关启用和禁用配置文件的背景信息，请参见[“使用基于配置文件的网络配置的准则” \[90\]](#)。

在启用和禁用各种类型的配置文件时，请参见以下附加准则：

- NCP – 在任何给定时刻，系统中必须有一个活动 NCP 和一个活动位置配置文件。活动 NCP 在您显式启用其他 NCP 之前一直保持活动状态。启用其他 NCP 将隐式禁用当前活动的 NCP。无法显式禁用系统上当前活动的 NCP。[“NCP 说明” \[87\]](#)。  
通过启用 `DefaultFixed` NCP 切换到固定模式时，`DefaultFixed` 位置也将自动启用，且无法更改。

启用 Automatic NCP 时，激活策略将根据当前网络条件选择合适的对应位置，然后启用该位置。

- NCU – 如果 NCU 的激活模式设置为 manual，则您可以手动启用和禁用属于当前活动 NCP 的单个 NCU。如果未指定链路或 NCU 类，则将启用或禁用所有 NCU。请参见“[NCU 说明](#)” [87]。
- 位置 – 缺省情况下，系统选择最佳位置配置文件来启用。系统从一组具有 system 或 conditional 激活模式的位置中选择位置。但是，无论位置的激活模式如何，用户随时都可以通过手动启用任何位置来覆盖系统的选择。手动启用位置时，系统将不会自动更改活动位置。将禁用自动选择位置的功能。必须显式禁用手动启用的位置才能恢复系统的条件位置选择。请参见“[位置配置文件说明](#)” [88]。
- ENM – 这些配置文件可以具有 manual 或 conditional 激活模式。如果将 activation-mode 属性设置为 conditional，则系统将基于指定的条件启用或禁用 ENM。如果将激活模式设置为 manual，则可以使用 netadm 命令启用或禁用 ENM。对 ENM 激活没有限制。在任何给定时刻，系统上可以有零个或多个 ENM 处于活动状态。启用或禁用一个 ENM 对当前处于活动状态的其他 ENM 没有任何影响。请参见“[配置文件激活模式](#)” [91]。

#### 例 13 启用 NCP

在以下示例中，启用了名为 myncp 的用户定义的 NCP。

```
$ netadm enable -p ncp myncp
Enabling ncp 'myncp'
```

#### 例 14 启用位置配置文件

在以下示例中，启用了名为 office 的位置配置文件。

```
$ netadm enable -p loc office
Enabling loc 'office'
```

请注意，指定配置文件名称时，netadm 命令不区分大小写。

#### 例 15 禁用链路 NCU

在以下示例中，禁用了一个名为 net1 的链路 NCU。

```
$ netadm disable -p ncu -c phys net1
```

#### 例 16 禁用位置

在以下示例中，禁用了名为 "office" 的位置。

```
$ netadm disable -p loc office
Disabling loc 'office'
```

### 例 17 启用和禁用 ENM

在以下示例中，启用了 ENM test-enm1 并禁用了另一个 ENM test-enm2。

```
$ netadm enable -p enm test-enm1
Enabling enm 'test-enm1'
$ netadm disable -p enm test-enm2
Disabling enm 'test-enm2'
```

### 例 18 在固定模式和反应性模式之间切换

以下示例说明如何通过启用系统定义的 DefaultFixed NCP 来切换到固定模式。

```
$ netadm enable -p ncp DefaultFixed
Enabling ncp 'DefaultFixed'
```

## 配置配置文件

可以采用以下方式使用 netcfg 命令来配置反应性配置文件：

- 交互
- 命令行模式
- 命令文件模式

## 在 netcfg 交互模式下工作

以交互模式使用时，对 netcfg 命令使用了范围的概念。以交互方式使用该命令时，您在任何给定时间所处的范围取决于配置文件类型以及您正在执行的特定任务。在终端窗口中键入 netcfg 命令本身时，如以下示例中所示，将在全局范围下显示一个提示符：

```
$ netcfg
netcfg>
```

要创建或选择配置文件，必须先启动 netcfg 交互式会话。

在全局范围提示符下，您可以使用 select 或 create 子命令查看、修改或创建以下顶级配置文件类型：

- NCP
- 位置
- ENM
- 已知 WLAN

在创建或选择配置文件之后，netcfg 交互式提示的语法类似于以下示例：

```
netcfg:object-type:object-name>
```

以交互模式使用 `netcfg` 命令可执行以下任务：

- 创建配置文件。
- 选择和修改配置文件。
- 验证关于配置文件的所有所需信息已设置且是有效的。
- 提交对新的配置文件的更改。
- 取消当前的配置文件配置，不将任何更改提交到持久性存储。
- 恢复对配置文件所做的更改。

对于位置配置文件和 ENM，在 `netcfg` 交互模式下选择或创建顶级配置文件将导致在配置文件范围下显示一个命令提示符，如以下示例中所示：

```
netcfg> select loc test-loc
netcfg:loc:test-loc>
```

当选择了某个 NCP 时，将在 NCP 范围下显示一个命令提示符。将在此范围内选择和创建 NCU。选择或创建 NCU 会将会话移动到所选 NCP 的配置文件范围内。在此范围内，可以查看并设置与当前选定的配置文件相关联的所有属性。

以下示例选中了 office NCP，这会将交互式会话移动到 NCP 的 NCP 范围，然后在该范围中选择 NCU。此操作将导致应用所选 NCU 的配置文件范围。在此范围内，可以查看或设置 NCU 的属性。

```
$ netcfg
netcfg> select ncp office
netcfg:ncp:office> select ncu phys net2
netcfg:ncp:office:ncu:net2>
```

在任何给定的范围内，命令提示符都指示当前选定的配置文件。在此范围内对配置文件所做的任何更改都会提交，这意味着更改将保存到持久性存储。退出范围时会隐式提交更改。如果不希望提交所做的更改，则可以恢复到该配置文件上次提交的状态。此操作将恢复对在这一级别的配置文件所做的任何更改。`revert` 和 `cancel` 子命令产生的效果相似。

有关说明，请参见“[创建 NCP](#)” [99]。

---

注 - `walkprop` 子命令用于显示单独与某个配置文件关联的各个属性。此命令只有在交互模式下使用时有意义。

---

## 在 `netcfg` 命令行模式下使用

在命令行模式下，任何影响选定配置文件或属性的 `netcfg` 子命令，都必须在该选定配置文件或属性所在的特定范围内执行。在命令行模式下选择一个配置文件与在交互模式下选择该文件相同。唯一的区别是在命令行模式下，所有子命令都在命令行上输入。例如，要列出某个 NCP 的内容，应先选择该 NCP，然后使用 `list` 子命令显示该 NCP 的 NCU，如以下示例中所示：

```
$ netcfg "select ncp myncp; list"
ncp:myncp
  management-type reactive
NCUs:
  phys net0
  phys net1
  ip net0
  ip net1
```

在此模式下使用 netcfg 命令时，请参见以下准则：

- 使用分号分隔各个子命令。
- 从全局范围指定 select 子命令以移动到 NCP 范围。
- 从 NCP 范围指定 list 子命令以列出该范围内的属性。
- 使用直双引号以防止 shell 解释分号。

---

注 - 在命令行模式下，必须在一行中键入完整的命令。使用 netcfg 命令在命令行模式下对所选配置文件所做的更改，在执行命令时将提交到持久性存储。

可以在命令行模式下使用任意 netcfg 子命令，不过 walkprop 子命令除外。

---

## 在 netcfg 命令文件模式下使用

在命令文件模式下，配置文件配置信息和命令是从文件中提取的。文件中的命令与在交互模式下使用的命令以及 export 子命令使用的命令相同。使用带有 -f 选项的 export 子命令生成文件。例如，以下命令将当前配置文件配置导出到一个文件：

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

要从文件导入配置文件配置，请键入以下命令：

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

export 子命令也能以交互方式使用。有关更多信息，请参见[“导出配置文件配置” \[117\]](#)。

## 创建 NCP

NCP 定义系统的网络配置。只能创建反应性 NCP，不能创建固定 NCP。

要在交互模式下创建 NCP，首先启动交互式会话。然后，使用 create 子命令创建新 NCP，如下示例中所示：

```
$ netcfg
netcfg> create ncp myncp
netcfg:ncp:myncp>
```

**例 19** 通过克隆 Automatic NCP 来创建 NCP

可以通过克隆除 DefaultFixed NCP 之外的任意现有 NCP 来创建 NCP。然后，修改该 NCP 的属性以指定新配置参数。在以下示例中，通过克隆系统定义的 Automatic NCP 创建了一个名为 newncp 的新 NCP。

```
$ netcfg
netcfg> create -t Automatic ncp newncp
netcfg:ncp:newncp> list
ncp:newncp
  management-type reactive
NCUs:
  phys net0
  phys net1
  ip net0
  ip net1
```

management-type 属性为只读属性，始终设置为 reactive。在上例中，list 子命令用于显示新复制的 NCP (newncp) 的内容。

还可以使用 netcfg 命令行模式来直接克隆 Automatic NCP，如以下示例中所示。交互式方法和命令行方法都会将现有 NCU 从 Automatic NCP 复制到新创建的 NCP。

```
$ netcfg create -t Automatic ncp newncp
$ netcfg list ncp newncp
ncp:newncp
  management-type reactive
NCUs:
  phys net0
  phys net1
  ip net0
  ip net1
```

## 为 NCP 创建 NCU

NCP 本质上是一个容器，包括配置有属性的一组 NCU，这些属性定义了 NCP 的网络配置。所有 NCP 都包含链路 NCU 和接口 NCU。

链路 NCU 指定 NCP 的链路配置和链路选择策略。接口 NCU 指定 NCP 的接口配置策略。如果要求 IP 连接，就需要提供链路 NCU 和接口 NCU 两者。必须使用 netcfg 命令或网络管理 GUI 显式添加或删除 NCU。有关使用网络管理 GUI 添加和删除 NCU 的更多信息，请参见“[从桌面管理网络配置](#)” [120]。

可以使用 netcfg 命令在交互模式或命令行模式下创建 NCU。因为创建 NCU 涉及到多个操作，所以在交互模式下创建 NCU，比尝试构建一行命令来创建 NCU 及其所有属性更为轻松高效。NCU 可以在最初创建 NCP 时创建，也可以在这之后创建。创建或修改 NCU 的过程包括设置常规的 NCU 属性，以及设置专门适用于每个 NCU 类型的属性。

当您以交互方式创建 NCU 时，`netcfg` 命令会遍历 NCU 的每个相关属性，显示其缺省值（如果有）和所有可能值。例如，如果为接口 NCU 的 `ipv4-addrsrc` 属性指定 `dhcp`，则不会提示您指定 `ipv4-addr` 属性的值，因为此属性仅用于配置静态 IP 地址。可以在交互式提示下为各个属性指定备用值。不指定值，按回车键，就会重新应用缺省值；如果没有缺省值，就会将该属性留空。

创建或修改 NCU 时，可以指定多个 NCU 属性。一些属性同时适用于两种 NCU 类型，而另外一些仅适用于链路 NCU 或接口 NCU。有关所有 NCU 属性（包括指定这些属性时可能适用的规则和条件）的完整说明，请参见 [netcfg\(1M\)](#) 手册页。

## ▼ 如何以交互方式为 NCP 创建 NCU

以下过程介绍了如何选择现有 NCP，然后以交互方式为该 NCP 创建 NCU。

---

注 - 在初始配置文件创建期间执行的“遍历”过程可以确保，在修改 NCP 时，系统根据您在创建期间所做的选择，仅向您提示那些适用的属性。

---

1. 启动 `netcfg` 交互式会话。

```
$ netcfg
netcfg>
```

2. 选择现有 NCP。

在以下示例中，选择了 NCP `myncp`：

```
netcfg> select ncp myncp
netcfg:ncp:myncp>
```

选择 NCP 会自动使您进入该 NCP 的范围。对于位置、ENM 或 WLAN 对象，命令提示符会使您进入该配置文件的配置文件范围。

3. 为 NCP 创建链路 NCU 和接口 NCU。

以下示例中创建了链路 NCU：

```
netcfg:ncp:myncp> create ncu phys net0
Created ncu `net0'. Walking properties ...
activation-mode (manual) [manual|prioritized]>
mac-address>
autopush>
mtu> 1600
```

其中 `ncu` 是对象类型，`phys` 是 NCU 的类，而 `net0` 是对象名称。

创建 NCU 会使您进入该对象的范围，并使您遍历该对象的缺省属性。

在本例中，指定了以下属性：

- `activation-mode` 属性缺省为 `manual`，按回车键即可接受。

- `mac-address` 和 `autopush` 属性将留空。
- `mtu` 属性设置为值 1600。

以下示例说明如何创建接口 NCU：

```
netcfg:ncp:myncp> create ncu ip net0
Created ncu `net0'. Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addrsrc (dhcp) [dhcp|static]> dhcp
ipv4-default-route>
```

其中 `ncu` 是对象类型，`ip` 是对象类，而 `net0` 是对象名称。

创建 NCU 会使您进入该对象的范围，并使您遍历该对象的缺省属性。

在本例中，指定了以下属性：

- `ip-version` 属性设置为 `ipv4`。
- `ipv4-addrsrc` 属性设置为 `dhcp`。

在创建 NCU 的过程中，`class` 选项用于区分 NCU 的两种类型。

4. (可选) 如下所示确保配置正确：

```
netcfg:ncp:myncp:ncu:net0> verify
All properties verified
```

`verify` 子命令验证配置并通知是否缺少任何必需值。

5. 保存创建的各个 NCU。

- 使用 `commit` 子命令：

```
netcfg:ncp:myncp:ncu:net0> commit
Committed changes
netcfg:ncp:myncp:ncu:net0>
```

`commit` 子命令会隐式验证属性。

- 使用 `end` 子命令：

```
netcfg:ncp:myncp:ncu:net0> end
Committed changes
netcfg:ncp:myncp>
```

`end` 子命令会隐式提交更改。

在本例中，如果已完成将 NCU 添加到 NCP，则 `end` 子命令会将会话移动到 NCP 范围。

在交互模式下，更改不会保存到持久性存储，直到您提交这些更改。使用 `commit` 子命令时，会提交整个配置文件。为了保持持久性存储的一致性，`commit` 操作还包括验证步

骤。如果验证失败，`commit` 操作也将失败。如果隐式提交失败，则会向您提供选项以结束或退出交互式会话，而不提交当前更改。也可以保持在当前范围内并继续对配置文件进行更改。

---

注 - 要取消所做的更改，可使用 `cancel` 或 `revert` 子命令。

`cancel` 子命令结束当前的配置文件配置而不将当前更改提交到持久性存储，然后将交互式会话上移到高一级的范围。`revert` 子命令撤消您进行的更改，并重新读取以前的配置。使用 `revert` 子命令时，交互式会话将保持在相同的范围内。

---

6. 完成后，退出交互式会话。

```
netcfg:ncp:myncp> exit
```

`exit` 子命令类似于 `end` 子命令，但也会退出交互式会话。

## 创建位置

位置配置文件包含的属性用于定义与基本链路和 IP 连接不直接相关的网络配置值。例如，在需要时结合应用的命名服务和 IP 过滤器设置。系统中始终必须有且只能有一个位置配置文件和一个 NCP 处于活动状态。

可以使用 `netcfg` 命令在交互模式或命令行模式下创建位置配置文件。创建位置配置文件时，通过指定那些在启用特定位置时定义特定配置参数的值，来设置位置的属性。位置属性按组进行分类，这里的组表示配置首选项的某个特定类。

位置属性也存储在系统信息库中。当启用某个特定位置时，其属性将自动应用到正在运行的系统。创建和修改位置配置文件还涉及到设置用于定义何时启用特定位置的属性。在配置过程中为您提供的属性基于以前设置的属性值。

有关所有位置配置文件属性（包括指定任意这些属性时可能适用的任何规则、条件和相关项）的完整说明，请参见 [netcfg\(1M\)](#) 手册页。

### ▼ 如何以交互方式创建位置

以下过程描述如何创建位置配置文件。在创建初始配置文件期间，所执行的“遍历”过程会根据您先前输入的值，仅向您提示那些适用的属性。

1. 启动 `netcfg` 交互式会话。

```
$ netcfg
netcfg>
```

2. 创建位置配置文件。

在以下示例中，创建了名为 office 的位置：

```
netcfg> create loc office
Created loc 'office'. Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-any
conditions> ncu ip:net0 is active
nameservices (dns) [dns|files|nis|ldap]>
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]>
nfsv4-domain>
ipfilter-config-file> /export/home/test/wifi.ipf.conf
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file> /etc/inet/ike/ikev1.config
ikev2-config-file>
ipsecpolicy-config-file>
```

创建位置会自动将您移到此位置的配置文件范围内。

在本例中，设置了以下属性：

- activation-mode 属性设置为 conditional-any，这会出现命令提示符，在其中启用了要指定的激活条件。
- 激活位置的条件指定为：ncu ip:net0 is active。
- 对于 ipfilter-config-file 属性，指定了 /export/home/test/wifi.ipf.conf 文件。
- 对于 ike-config-file 属性，指定了 /etc/inet/ike/ikev1.config 文件。
- 对于其余属性，通过按回车键接受缺省值。

3. (可选) 使用 list 子命令显示配置文件配置，如下所示：

```
netcfg:loc:office> list
loc:office
      activation-mode          conditional-any
      conditions                "ncu ip:net0 is active"
      enabled                   false
      nameservices              dns
      nameservices-config-file  "/etc/nsswitch.dns"
      dns-nameservice-configsrc dhcp
      ipfilter-config-file      "/export/home/test/wifi.ipf.conf"
      ike-config-file           "/etc/inet/ike/ikev1.config"
```

4. 验证配置文件配置是正确的。

```
netcfg:loc:office> verify
All properties verified
```

verify 子命令会验证您的配置并通知是否缺少任何必需值。

5. 验证配置之后，保存位置。

```
netcfg:loc:office> commit
```

```
Committed changes
```

另外，还可以使用 `end` 子命令结束会话，这样也会保存配置文件配置并将会话移动到全局范围。

```
netcfg:loc:office> end
Committed changes
netcfg>
```

## 6. 退出交互式会话。

```
netcfg> exit
```

## 创建 ENM

通过 ENM 可以指定应用程序或脚本（例如 VPN 应用程序）应何时执行网络配置，对于在 NCP 和位置配置文件中指定的配置来说，这是独立的（外部）网络配置。有关 ENM 的更多信息，请参见“[ENM 说明](#)” [88]。

---

注 - 系统不会自动识别您可能要为其创建 ENM 的应用程序。必须先在系统上安装和配置这些应用程序，然后再使用 `netcfg` 命令为其创建 ENM。

---

有关在创建 ENM 时可指定的属性的更多信息，请参见 [netcfg\(1M\)](#) 手册页。

## ▼ 如何以交互方式创建 ENM

### 1. 启动 `netcfg` 交互式会话。

```
$ netcfg
netcfg>
```

### 2. 创建 ENM。

```
netcfg> create enm 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>
```

创建 ENM 会自动将您移动到 ENM 的配置文件范围内并遍历其各个属性。

在此示例中，为 `test-enm` ENM 指定了以下属性：

- `activation-mode` 属性，该属性设置为手动，按回车键即可接受。由于此值设置为 `manual`，因此无法设置 `conditions` 属性。

- fmri 属性设置为 `svc:/application/test-enm:default`。
- 没有为此 ENM 设置 `start` 和 `stop` 属性。

3. (可选) 显示配置文件配置。

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

4. 验证配置文件配置是正确的。

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

`verify` 子命令会验证您的配置并通知是否缺少任何必需值。

5. 保存 ENM。

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

`commit` 子命令验证并保存配置。  
也可以使用 `end` 子命令结束会话，这样也会保存配置文件配置。

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

6. 退出交互式会话。

```
netcfg> exit
```

## 创建已知 WLAN

已知 WLAN 是存储系统已知的无线网络的相关信息的配置文件。NCP 可以基于配置信息自动配置无线接口，这些信息由系统所连接到的各个无线网络提供。有关更多信息，请参见“[已知 WLAN 的说明](#)” [89]。

有关在创建和修改 WLAN 时可指定的属性的信息，请参见 [netcfg\(1M\)](#) 手册页。

### ▼ 如何以交互方式创建已知 WLAN

1. 启动 `netcfg` 交互式会话。

```
$ netcfg
```

```
netcfg>
```

## 2. 创建已知 WLAN。

以下示例创建连接到名为 ESSID 的无线网络的已知 WLAN。已知 WLAN 必须与无线网络名称或其 ESSID 具有相同名称：

```
netcfg> create wlan mywifi
Created wlan 'mywifi'. Walking properties ...
priority (0)> 100
bssids>
keyname> mywifi-key
keyslot>
security-mode [none|wep|wpa]> wpa
netcfg:wlan:mywifi>
```

创建 WLAN 会自动将您移动到 WLAN 的配置文件范围内并遍历其各个属性。

在此示例中，为 mywifi 已知 WLAN 指定了以下属性：

- priority 属性的值从缺省值 0 更改为 100。
- keyname 属性设置为 mywifi-key 并指定此无线网络的安全对象名称。有关更多信息，请参见“[建立安全的 WiFi 通信](#)” [128]。
- security-mode 属性设置为 wpa。此属性指定该无线网络使用的加密类型。
- keyslot 和 bssid 属性值将留空。

## 3. (可选) 显示配置文件配置。

```
netcfg:wlan:mywifi> list
known wlan:mywifi
priority 100
keyname "mywifi-key"
security-mode wpa
netcfg:wlan:mywifi>
```

## 4. 验证配置文件配置是正确的。

```
netcfg:wlan:mywifi> verify
All properties verified
```

verify 子命令会验证您的配置并通知是否缺少任何必需值。

## 5. 保存已知 WLAN。

```
netcfg:wlan:mywifi> commit
Committed changes
```

commit 子命令验证并保存配置。

另外，还可以使用 end 子命令结束会话，这样也会保存配置文件配置并将会话移动到全局范围。

```
netcfg:wlan:mywifi> end
```

```
Committed changes
netcfg>
```

6. 退出交互式会话。

```
netcfg> exit
```

## 管理配置文件

本节包含以下主题：

- “设置配置文件的属性值” [108]
- “获取有关配置文件配置的信息” [110]
- “使用 walkprop 子命令为配置文件设置属性值” [113]
- “显示有关配置文件的信息” [115]
- “删除配置文件” [116]
- “导出配置文件配置” [117]
- “恢复导出的配置文件配置” [119]

## 设置配置文件的属性值

使用 set 子命令设置或修改反应性配置文件的属性值。此子命令可以在交互模式或命令行模式下使用。如果在命令行模式下设置或更改属性值，更改将立即提交到持久性存储。

---

注 - 不能使用 set 子命令修改 DefaultFixed NCP 或 DefaultFixed 位置配置文件。只要 DefaultFixed NCP 处于活动状态，就可使用 dladm 和 ipadm 命令进行配置更改。DefaultFixed 位置处于活动状态时，使用 svccfg 和 svcadm 命令直接对相关 SMF 属性进行更改。请参见 [svccfg\(1M\)](#) 和 [svcadm\(1M\)](#) 手册页。

---

set 子命令的语法如下所示：

```
netcfg> set prop-name=value1[,value2,...]
```

### ▼ 如何以交互方式设置配置文件属性值

以下过程描述如何以交互方式为位置配置文件设置属性值。以交互方式设置属性值时，必须先从前范围选择一个配置文件，这会将交互式会话移到该配置文件范围内。选定的配置文件随后从持久性存储加载到内存中。然后，在此范围内，可以修改配置文件的属性。

以下过程仅用于示范目的，说明了如何以交互方式设置 test-loc 位置的 ipfilter-config-file 属性。

1. 启动 netcfg 交互式会话。

```
$ netcfg
netcfg>
```

2. 选择要修改的配置文件或配置对象。

```
netcfg> select loc test-loc
netcfg:loc:test-loc>
```

3. 设置属性值。

在以下示例中，设置 ipfilter-config-file 属性：

```
netcfg:loc:test-loc> set ipfilter-config-file = /path/to/ipf-file
```

4. (可选) 列出配置信息。

```
netcfg:loc:test-loc> list
loc:test-loc
activation-mode  manual
enabled          false
nameservices    dns
dns-nameservice-configsrc  dhcp
nameservices-config-file  "/etc/nsswitch.dns"
ipfilter-config-file  "/path/to/ipf-file"
```

5. 结束会话。

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

end 子命令保存会话并将其移动到全局范围。

6. 退出交互式会话。

```
netcfg> exit
```

例 20 在命令行模式下为配置文件设置属性值

以上示例说明如何以交互方式设置 ipfilter-config-file 属性，该操作还可在命令行模式下执行，如下所示：

```
$ netcfg "select loc test-loc; set ipfilter-config-file = /path/to/ipf-file"
```

在只需要执行简单操作时，命令行模式最为合适。但是，通过在命令行上仔细指定合适的子命令，还可以使用命令行模式来执行更复杂的操作。对于交互式示例，在命令行模

式下，还必须先选择位置以移动到该配置文件范围中。然后，可以指定 `set` 子命令来设置单独的属性值。

使用命令行模式时，可以为给定的属性同时设置多个值。使用此方法设置多个值时，各个值之间必须用逗号 (,) 分隔。如果指定属性的各值中包含逗号，作为属性值一部分的逗号前面就必须加反斜杠 (\)。包含在只有一个值的属性内的逗号不解释为分隔符，因此不需要在逗号前面加反斜杠。

例如，如下所示设置 `ip-version` 属性，为 `myncp` NCP 中的 `net0` NCU 同时使用 IPv4 和 IPv6：

```
$ netcfg "select ncp myncp; select ncu ip net0; set ip-version=ipv4,ipv6"
```

## 获取有关配置文件配置的信息

可以在交互模式或命令行模式下，使用 `list` 子命令列出所有配置文件、属性值对以及当前范围或指定范围中存在的资源。

在全局范围中，`list` 子命令列出系统中所有的系统定义的配置文件和用户定义的配置文件，如以下示例中所示：

```
$ netcfg list
netcfg list
NCPs:
  DefaultFixed
  Automatic
  myncp
Locations:
  Automatic
  NoNet
  DefaultFixed
  office
ENMs:
  test-enm
WLANs:
  mywifi
```

---

注 - 在交互模式下，在全局范围内使用 `list` 子命令会列出相同的信息。

---

请注意，`list` 子命令不会列出各个配置文件的状况。要显示有关配置文件的状况及其状态，请使用 `netadm` 命令和 `list` 子命令。有关更多信息，请参见[“显示配置文件的当前状态” \[115\]](#)。

## 列出单个配置文件的属性值

配置文件范围中的 `list` 子命令列出指定配置文件的所有属性值。其语法如下所示：

```
netcfg> list [ object-type [ class ] object-name ]
```

## ▼ 如何以交互模式列出某个配置文件的所有属性值

以下过程介绍如何使用 `list` 子命令以交互模式列出某个配置文件的所有属性值。以下过程中的示例说明如何列出名为 `net0` 的 IP NCP 的配置信息。为每个配置文件列出的值会根据配置文件而不同。

1. 启动 `netcfg` 交互式会话。

```
$ netcfg
netcfg>
```

2. 选择 NCP。

```
netcfg> select ncp myncp
netcfg:ncp:myncp>
```

3. 使用以下方法之一列出 IP NCU 的配置：

- 从全局范围列出配置：

```
netcfg:ncp:myncp> list ncu ip net0
ncu:net0
      type           interface
      class          ip
      parent         "myncp"
      enabled        false
      ip-version     ipv4,ipv6
      ipv4-addrsrc   dhcp
      ipv6-addrsrc   dhcp,autoconf
netcfg:ncp:myncp>
```

- 从配置文件范围列出配置：

```
netcfg:ncp:myncp> select ncu ip net0
netcfg:ncp:myncp:ncu:net0> list
ncu:net0
      type           interface
      class          ip
      parent         "myncp"
      enabled        false
      ip-version     ipv4,ipv6
      ipv4-addrsrc   dhcp
      ipv6-addrsrc   dhcp,autoconf
netcfg:ncp:myncp:ncu:net0>
```

4. 退出交互式会话。

```
netcfg:ncp:myncp:ncu:net0> exit
```

例 21 以命令行模式列出属性值

上例说明如何以交互方式列出属性值，该操作还能够在命令行模式下执行。不论使用什么模式，输出都是相同的。

例如，如下所示列出 NCP 范围中的 net0 IP NCU 的属性：

```
$ netcfg "select ncp myncp; list ncu ip net0"
```

要列出配置文件范围中 net0 IP NCU 的属性，可使用以下命令：

```
$ netcfg "select ncp myncp; select ncu ip net0; list"
```

## 获取配置文件的特定属性值

使用 get 子命令获取配置文件的特定属性值。此子命令可以在交互模式或命令行模式下使用。该命令语法如下所示：

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

### ▼ 如何获取配置文件的特定属性值

以下过程介绍如何使用 get 子命令以交互模式获取某个配置文件的特定属性值。以下过程中的示例说明如何获取 IP NCU 的 ip-version 属性。

1. 启动 netcfg 交互式会话。

```
$ netcfg
netcfg>
```

2. 选择 NCP，然后选择 IP NCU。例如：

```
netcfg> select ncp myncp
netcfg:ncp:myncp> select ncu ip net0
netcfg:ncp:myncp:ncu:net0>
```

3. 使用以下命令之一获取特定属性值：

- 使用 get 子命令可显示属性名称和属性值，如下所示：

```
netcfg:ncp:myncp:ncu:net0> get ip-version
ip-version ipv4,ipv6
```

- 如果仅希望获取属性值而不显示属性名称，请如下所示在 get 子命令中使用 -v 选项：

```
netcfg:ncp:myncp:ncu:net0> get -v ip-version
ipv4,ipv6
```

#### 4. 退出交互式会话。

```
netcfg:ncp:myncp:ncu:net0> exit
```

#### 例 22 在命令行模式下获取配置文件的特定属性值

以上交互模式示例也可以在命令行模式下执行。不论使用什么模式，输出都是相同的。

例如，可以如下所示，在命令行模式下获取 IP NCU 的 `ip-version` 属性的值：

```
$ netcfg "select ncp myncp; select ncu ip net0; get ip-version"
ip-version ipv4,ipv6
```

以下示例说明如何使用 `get` 子命令和 `-v` 选项来获取特定属性值。此方法对于不需要解析属性名称的脚本非常有用。

```
$ netcfg "select ncp myncp; select ncu ip net0; get -V ip-version"
ipv4,ipv6
```

## 使用 `walkprop` 子命令为配置文件设置属性值

使用 `walkprop` 子命令以交互方式查看和更改配置文件的单个属性值。启动交互式会话之后，可以键入 `walkprop` 子命令以显示配置文件的各个属性的名称和当前值，一次显示一个属性。在查看各个属性时，可以根据需要设置或更改当前值或缺省值。

---

注 - `walkprop` 子命令意味着仅在交互模式下使用。

---

### ▼ 如何“遍历”和设置指定配置文件的属性值

以下过程介绍了如何使用 `walkprop` 子命令以交互模式查看和更改指定配置文件的属性值。如以下示例中所示，使用 `walkprop` 子命令设置配置文件的属性时，无需使用 `set` 子命令。

#### 1. 启动 `netcfg` 交互式会话。

```
$ netcfg
netcfg>
```

#### 2. 选择要查看和更改其属性的配置文件。

在以下示例中，选择了名为 `test-loc` 的位置：

```
netcfg> select loc test-loc
netcfg:loc:test-loc>
```

#### 3. 键入 `walkprop` 子命令以开始遍历。

在以下示例中，发布 walkprop 子命令之后，显示的第一个属性是 activation-mode 属性。请注意，属性的缺省值当前设置为 manual（如括号中所示）。

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

4. 要修改属性值，请在交互式提示下键入新值，然后按回车键。  
例如，可以如下所示将位置的 activation-mode 属性从 manual 更改为 conditional-all：

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

按回车键可保存当前设置并转到下一个属性。

5. 重复遍历过程，直至该配置文件的所有属性都已显示，根据需要按照第 4 步中的说明进行修改。

```
netcfg:loc:test-loc> walkprop
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
conditions> advertised-domain is example.com
nameservices (dns) [dns|files|nis|ldap]>
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]>
nfsv4-domain>
ipfilter-config-file>
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ikev2-config-file>
ipsecpolicy-config-file>
```

不进行任何更改按回车键可保留所有现有缺省值，然后继续遍历到下一个属性。

---

注 - 只显示指定配置文件的相关属性，如[如何以交互方式创建位置 \[103\]](#)中所述。

---

6. 列出配置文件的当前缺省属性值。例如：

```
netcfg:loc:test-loc> list
loc:test-loc
activation-mode    conditional-all
conditions        "advertised-domain is example.com"
enabled           false
nameservices      dns
nameservices-config-file  "/etc/nsswitch.dns"
dns-nameservice-configsrc  dhcp
```

请注意，在前面的输出中，activation-mode 属性现在设置为 conditional-all。

7. 退出交互式会话以提交更改。

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

## 显示有关配置文件的信息

可以使用 `netadm` 命令和 `list` 子命令显示系统上任意或全部配置文件的相关信息，包括各个配置文件的当前状态。

`list` 子命令的语法如下：

```
$ netadm list [ -p object-type ] [ -c ncu-class ] [ object-name ]
```

如下所示显示系统上的所有配置文件以及各个配置文件的当前状态：

```
$ netadm list
TYPE  PROFILE  STATE
ncp   DefaultFixed  disabled
ncp   Automatic  online
ncu:phys net0    online
ncu:phys net1    offline
ncu:ip  net0    online
ncu:ip  net1    offline
loc   Automatic  online
loc   NoNet     offline
loc   DefaultFixed  offline
enm   test-enm  disabled
```

请注意，`list` 子命令显示已启用的 NCP 和构成该特定 NCP 的所有 NCU。

## 显示配置文件的当前状态

可以在 `list` 子命令中包括配置文件类型和 NCU 类来标识特定配置文件。如果只提供一个配置文件类型，则显示该类型的所有配置文件。如果按其名称指定配置文件，则显示该配置文件的当前状态。如果配置文件名称不是唯一的，会列出具有该名称的所有配置文件。

例 23 显示指定的配置文件的当前状态

以下示例列出名为 `Automatic` 的系统上所有配置文件的当前状态。

```
$ netadm list Automatic
TYPE  PROFILE  STATE
ncp   Automatic  online
ncu:ip  net1    offline
ncu:phys net1    offline
ncu:ip  net0    online
ncu:phys net0    online
loc   Automatic  online
```

在以下示例中，组合使用了 `list` 子命令和 `-p` 选项来显示当前在系统上的所有位置。在特定示例中，**Automatic** 位置处于联机状态（已启用）。

```
$ netadm list -p loc
TYPE PROFILE STATE
loc DefaultFixed offline
loc NoNet offline
loc Automatic online
```

在以下示例中，组合使用了 `list` 子命令和 `-c` 选项来显示活动 NCP 中的所有接口 NCU。

```
$ netadm list -c ip
TYPE PROFILE STATE
ncu:ip net0 online
ncu:ip net1 offline
```

## 显示配置文件的辅助状态值

配置文件的辅助状态解释了为什么给定的配置文件处于联机还是脱机状态（启用或禁用）。要列出辅助状态值，请结合使用 `-x` 选项和 `list` 子命令，如下所示：

```
$ netadm list -x
TYPE PROFILE STATE AUXILIARY STATE
ncp DefaultFixed disabled disabled by administrator
ncp Automatic online active
ncu:phys net0 online interface/link is up
ncu:phys net1 offline interface/link is down
ncu:ip net0 online interface/link is up
ncu:ip net1 offline conditions for activation are unmet
loc Automatic offline conditions for activation are unmet
loc NoNet offline conditions for activation are unmet
enm test-enm disabled disabled by administrator
loc DefaultFixed offline conditions for activation are unmet
```

辅助状态值因具体配置文件类型而异。有关辅助状态的更多信息，请参见 [nwamd\(1M\)](#) 手册页。

## 删除配置文件

可以使用 `destroy` 子命令删除用户定义的配置文件和配置对象，例如 NCU。不能删除系统定义的配置文件，包括：**Automatic** 和 **DefaultFixed** NCP，以及 **Automatic**、**NoNet** 和 **DefaultFixed** 位置。另请注意，不能删除当前活动的配置文件。必须先禁用配置文件，然后才能删除。

`destroy` 子命令的语法如下所示：

```
netcfg> destroy [ -a | object-type [ class ] object-name ]
```

-a 选项从系统中删除所有用户定义的配置文件，不过当前活动的用户定义配置文件除外。

## ▼ 如何以交互方式删除配置文件

以下过程介绍如何以交互方式删除用户定义的配置文件。此过程仅用于示范目的，说明了如何从 myncp 用户定义的 NCP 中删除 IP NCU。

1. 启动 netcfg 交互式会话。

```
$ netcfg
netcfg>
```

2. 选择配置文件。

例如，要选择 myncp NCP，请键入以下命令：

```
netcfg> select ncp myncp
netcfg:ncp:myncp>
```

3. 删除配置文件或配置对象。

在以下示例中，从 myncp NCP 中删除 net1 IP NCU：

```
netcfg:ncp:myncp> destroy ncu ip net1
netcfg:ncp:myncp>
```

4. 退出交互式会话。例如：

```
netcfg:ncp:myncp> exit
```

例 24 在命令行模式下删除配置文件

以上示例说明如何以交互方式删除配置文件，该操作还可在命令行模式下执行，如下所示：

```
$ netcfg "select ncp myncp; destroy ncu ip net1"
```

## 导出配置文件配置

使用 export 子命令保存配置文件配置。如果您负责维护要求使用相同网络配置的多个服务器，则导出配置文件非常有用。export 子命令可在交互模式或命令行模式下使用。导出配置文件时，输出显示为 netcfg 命令可以解释的一系列子命令。这些子命令类似于在交互模式或命令行模式下键入的命令。

---

注 - 某些配置会限制 export 功能的使用。只能导出最初使用 netcfg 命令创建的配置对象。不能导出使用 dladm 或 ipadm 命令创建的配置对象，例如链路聚合或 IPMP 组。此外，不能导出 DefaultFixed NCP 和 DefaultFixed 位置。

---

export 子命令的语法如下所示：

```
netcfg> export [ -d ] [ -f output-file ] [ object-type [ class ] object-name ]
```

---

注 - export 子命令的 -d 和 -f 选项可以相互独立使用。-f 选项将当前范围内或指定范围内的当前配置输出到指定的文件。-d 选项添加 destroy -a 命令作为首行输出。

---

#### 例 25 以交互方式导出配置文件配置

以下示例说明如何使用 export 子命令以交互方式在屏幕上显示配置文件配置。

```
$ netcfg
netcfg> export
create ncp "myncp"
create ncu ip "net0"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
set ipv6-addrsrc=dhcp,autoconf
end
create ncu phys "net0"
set activation-mode>manual
set mtu=5000
end
end
create loc "test-loc"
set activation-mode=conditional-all
set conditions="system-domain is example.com"
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dhcp
end
create enm "test-enm"
set activation-mode=conditional-all
set conditions="ip-address is-not-in-range 10.2.3.4"
set fmri="svc:/application/test-enm:default"
end
create wlan "mywifi"
set priority=100
set keyname="mywifi-key"
set security-mode=wpa
end
```

在命令行模式下，键入以下命令：

```
$ netcfg export
```

可以将 `-d` 选项与 `export` 子命令结合使用，来添加 `destroy -a` 命令作为 `netcfg export` 输出的第一行，如下示例中所示（为简便起见，内容已截断）：

```
$ netcfg
netcfg> export -d
destroy -a
create ncp "myncp"
create ncu ip "net0"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
.
.
.
```

在命令行模式下，键入以下命令：

```
$ netcfg export -d
```

例 26 将配置文件配置导出到文件

在以下示例中，使用带有 `-f` 选项的 `export` 子命令，将 `myncp` NCP 的配置信息写入到一个文件中。在以下示例中，`-f` 选项将输出写入到一个名为 `myncp2` 的新文件中。`-d` 选项用于将 `destroy -a` 命令添加为 `netcfg export` 输出的第一行。

如下所示在交互模式下将配置文件配置导出到文件中：

```
$ netcfg
netcfg> export -d -f myncp2
```

如下所示在命令行模式下执行相同的任务：

```
$ netcfg export -d -f myncp2
```

以下截断的示例说明了如何显示配置文件配置：

```
$ cat myncp2
destroy -a
create ncp "myncp"
create ncu ip "net0"
.
.
.
```

## 恢复导出的配置文件配置

可以将使用 `export` 子命令生成的配置文件配置导入回系统中。导出的文件包含 `netcfg` 命令可以解释的一系列子命令。如果您需要恢复配置文件配置或者将配置文件配置复制

到其他系统，则 `export` 子命令非常有用。请注意，只能使用 `netcfg` 命令行模式恢复配置文件配置。

可以使用以下命令恢复导出的配置：

```
$ netcfg -f file
```

例如，如下所示恢复在例 26 “将配置文件配置导出到文件”中导出的文件：

```
$ netcfg -f myncp2
Configuration read.
```

## 从桌面管理网络配置

通过使用网络管理 GUI（以前称为 NWAM）可以从桌面管理网络配置。此工具类似于使用 `netcfg` 和 `netadm` 命令。利用该 GUI，可以连接到有线或无线网络，配置新的有线或无线连接，创建位置配置文件以及激活或取消激活配置文件。从桌面管理网络配置最适合笔记本个人电脑用户，以及网络条件经常发生变化的情况，例如，从家庭办公切换到工作场所的无线网络的情况或者出差的情况。

有关与从台式机管理无线网络的任务相关的信息，请参见“[从桌面管理无线网络](#)” [130]。

从桌面管理网络配置时，请遵循以下一般原则和最佳做法：

- 在从桌面管理网络配置时，最简单的解决方案是启用系统定义的 Automatic NCP。在家中时，可以使用此 NCP 连接到您家中的无线网络。
- 如果您决定使用有线连接，则可以插入以太网电缆。不要将缺省的 Automatic NCP 切换到其他 NCP。网络连接会自动从无线网络连接适应有线网络连接，无需对现有的网络配置进行任何其他更改。
- 在办公室，适用同样的规则。如果未将以太网电缆插入网络，则会启用 Automatic NCP，使用反应性网络，并自动建立无线网络连接。
- 如果 DefaultFixed NCP 当前处于活动状态，则只能查看其状态。要在此 NCP 处于活动状态时配置网络，必须使用 `dladm` 和 `ipadm` 命令。请参见第 2 章在 Oracle Solaris 中管理数据链路配置和第 3 章在 Oracle Solaris 中配置和管理 IP 接口和地址。
- 请记住，不管是家庭环境还是办公室环境，最初都必须选择一个无线网络，并将其保存到收藏的无线网络列表中（如果您尚未这样做的话）。

通过使用网络管理 GUI 或者运行 `netadm select-wifi` 命令，选择一个无线网络。请参见“[在反应性模式下管理已知 WLAN](#)” [130]。

- 要查看当前网络连接的状态，可将鼠标悬停在桌面上的网络状态通知图标上，或者直接单击该图标。该网络状态通知图标还包括上下文菜单，用于通过 GUI 创建和管理网络配置。

如果桌面上未显示网络状态通知图标，可通过选择 "System"（系统）-> "Administration"（管理）-> "Network"（网络）启动该图标。要从命令行启动 GUI，

请运行 `nwam-manager` 命令。有关更多信息，请参见 JDS/GNOME 手册页集中的 `nwam-manager(1M)` 手册页。

- 可在 "Network Preferences" (网络首选项) 对话框的 "Network Profile" (网络配置文件) 部分管理与 IP 相关的配置。通过单击桌面上的网络状态通知图标，或者在网络状态通知图标的上下文菜单中选择 "Network Preferences" (网络首选项) 选项，可以访问 "Network Preferences" (网络首选项) 对话框。



## 在 Oracle Solaris 中管理无线网络

---

本章介绍了用于在 Oracle Solaris 发行版中管理无线网络的任務。

IEEE 802.11 规范定义了局域网的无线通信。这些规范及其描述的网络统称为 *WiFi*，该术语是 Wi-Fi 联盟贸易组注册的商标。提供商和潜在客户都可以轻松配置 WiFi 网络。因此，WiFi 网络日益普及，在世界各地广泛使用。WiFi 网络与手提电话、电视和收音机使用相同的无线电波技术。

---

注 - Oracle Solaris 未包含配置 WiFi 服务器或访问点的功能。

---

本章包含以下主题：

- “使用命令行管理无线网络” [123]
- “建立安全的 WiFi 通信” [128]
- “在反应性模式下管理已知 WLAN” [130]
- “从桌面管理无线网络” [130]

### 使用命令行管理无线网络

本节介绍了以下任务：

- [如何连接到 WiFi 网络](#) [123]
- [如何监视 WiFi 链路](#) [127]

#### ▼ 如何连接到 WiFi 网络

开始之前 执行以下步骤将笔记本个人电脑连接到 WiFi 网络。

1. 成为管理员。
2. 显示系统上数据链路的物理属性。

```
# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX  DEVICE
```

```
net0          Ethernet          up      1500  full   ath0
net1          Ethernet          up      1500  full   e1000g0
```

在以上示例中，输出指示有两个链路可用。设备 ath0 链路上的 net0 支持 WiFi 通信。使用 e1000g0 链路可以将系统连接到有线网络。

### 3. 配置 WiFi 接口。

- a. 创建支持 WiFi 的接口。例如：

```
# ipadm create-ip net0
```

- b. 验证链路已激活。

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

### 4. 检查可用网络。

```
# dladm scan-wifi
LINK        ESSID      BSSID/IBSSID  SEC      STRENGTH  MODE  SPEED
net0        ofc        00:0e:38:49:01:d0  none    good      g     54Mb
net0        home      00:0e:38:49:02:f0  none    very weak g     54Mb
net0        linksys   00:0d:ed:a5:47:e0  none    very good g     54Mb
```

scan-wifi 命令显示有关在当前位置可用的 WiFi 网络的信息。输出包括以下信息：

LINK	请参阅 WiFi 连接中使用的链路名称。
ESSID	请参见“扩展服务集 ID”。ESSID 是 WiFi 网络的名称，该名称可由特定无线网络的管理员随机指定。
BSSID/IBSSID	请参见“基本服务集 ID”(Basic Service Set ID, ESSID)，这是针对特定 ESSID 的唯一标识符。BSSID 是服务于具有特定 ESSID 的网络的附近访问点的 48 位 MAC 地址。
SEC	请参见访问无线网络所需的安全性类型。值为 none、WEP 和 WPA。有关更多信息，请参见 <a href="#">“建立安全的 WiFi 通信” [128]</a> 。
STRENGTH	请参阅您位置处可用的 WiFi 网络发出的无线电信号的强度。
MODE	请参阅网络运行的 802.11 协议的版本。模式为 a、b、g 以及这些模式的任意组合。
SPEED	请参见特定网络的速度（以兆位/秒为单位）。

### 5. 使用以下方法之一连接 WiFi 网络：

- 连接到信号最强的不安全的 WiFi 网络。

```
# dladm connect-wifi
```

- 通过指定其 ESSID 连接到不安全的网络。

```
# dladm connect-wifi -e ESSID
```

有关使用 `dladm connect-wifi` 命令的更多信息，请参见“[建立安全的 WiFi 通信](#)” [128]和 `dladm(1M)` 手册页。

6. 如下所示检查系统所连接到的 WiFi 网络的状态：

```
# dladm show-wifi
LINK      STATUS      ESSID      SEC      STRENGTH  MODE  SPEED
net0      connected   ofc        none     very good g      36Mb
```

以上输出指示系统已连接到 `ofc` 网络。此过程中步骤 4 的 `scan-wifi` 输出指示 `ofc` 在可用网络中信号最强。`dladm connect-wifi` 命令会自动选择信号最强的 WiFi 网络，除非您显式指定其他无线网络。

7. 使用以下方法之一为接口配置 IP 地址：

- 从 DHCP 服务器获取 IP 地址。

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

如果 WiFi 网络不支持 DHCP，则将显示以下消息：

```
ipadm: interface: interface does not exist or cannot be managed using DHCP
```

- 配置静态 IP 地址。

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

如果您的系统有专用的 IP 地址，请使用此选项。

8. 使用以下方式之一，通过 WiFi 网络访问 Internet：

- 如果访问点提供免费服务，则可以运行浏览器或您选择的任何应用程序。
- 如果访问点位于需要付费的商业 WiFi 网络中，请遵循该位置提供的说明进行操作。通常需要为此选项提供密钥和付费方法。

9. 采用以下方法之一结束会话：

- 终止 WiFi 会话，但保持系统运行。

```
# dladm disconnect-wifi
```

- 当前正在运行多个会话时，终止特定的 WiFi 会话。

```
# dladm disconnect-wifi link
```

其中，*link* 代表要用于会话的接口。

- 在 WiFi 会话正在运行时，正常关闭系统。

```
# shutdown -g0 -i5
```

关闭系统之前，无需显式断开 WiFi 会话的连接。

#### 例 27 连接到特定 WiFi 网络

以下示例组合了要将 Oracle Solaris 系统连接到无线网络所需执行的不同步骤。该示例还显示了如何强制系统连接到特定的首选无线网络，而不是允许 OS 随机选择无线网络。在以下示例中，假定为您指定了在笔记本个人电脑上使用静态 IP 地址 10.192.16.3/24。

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

# ipadm create-ip net0
# ipadm show-if net0
IFNAME  CLASS  STATE  ACTIVE  OVER
lo0     loopback  ok     yes     --
net0    ip       ok     yes     --

# dladm scan-wifi
LINK  ESSID  BSSID/IBSSID  SEC  STRENGTH  MODE  SPEED
net0  wifi-a  00:0e:38:49:01:d0  none  weak      g     54Mb
net0  wifi-b  00:0e:38:49:02:f0  none  very weak g     54Mb
net0  ofc-net 00:0d:ed:a5:47:e0  wep   very good g     54Mb
net0  citinet 00:40:96:2a:56:b5  none  good      b     11Mb

# dladm connect-wifi -e citinet
# dladm show-wifi
LINK  STATUS  ESSID  SEC  STRENGTH  MODE  SPEED
net0  connected  citinet  none  good      g     11Mb

# ipadm create-addr -a 10.192.16.3/24 net0
ipadm: net0/v4
# ipadm show-addr net0
ADDROBJ  TYPE  STATE  ADDR
net0/v4  static  ok     10.192.16.3/24
```

启动浏览器或其他应用程序，开始通过 WiFi 网络工作。

```
# firefox
```

终止会话，但使个人电脑保持运行状态。

```
# dladm disconnect-wifi
# dladm show-wifi
LINK      STATUS      ESSID      SEC      STRENGTH  MODE      SPEED
net0      disconnected --         --       --        --        --
```

show-wifi 的输出确认您已从 WiFi 网络断开 net0 链路的连接。

## ▼ 如何监视 WiFi 链路

以下过程介绍如何通过标准网络工具监视 WiFi 链路的状态以及如何更改选定链路属性。

1. 成为管理员。
2. 连接到 WiFi 网络，如[如何连接到 WiFi 网络 \[123\]](#)中所述。
3. 查看系统的数据链路的属性。

```
# dladm show-linkprop link
```

4. 为链路设置固定的速度。



注意 - Oracle Solaris 自动为 WiFi 连接选择最佳速度。修改链路的初始速度可能会降低性能，或导致无法建立特定 WiFi 连接。

可以将链路速度修改为 show-linkprop 子命令输出中列出的可能值之一，如以下示例中所示：

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

5. 检查链路上的包流。

```
# netstat -I net0 -i 5
input net0      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
```

## 例 28 设置链路的速度

以下示例说明在连接到 WiFi 网络之后如何设置链路的速度。

```
# dladm show-linkprop -p speed net0
LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
net0      speed         r- 25           0              0
1,2,5,6,9,11,12,18,24,36,48,54
# dladm set-linkprop -p speed=36 net0

# dladm show-linkprop -p speed net0
LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
net0      speed         r- 36           0              0
1,2,5,6,9,11,12,18,24,36,48,54
```

## 建立安全的 WiFi 通信

无线电波技术使 WiFi 网络随时可用，而且经常供用户免费访问。因此，连接到 WiFi 网络可能会带来安全风险。

以下类型的 WiFi 连接更为安全：

- 连接到专用的、限制访问的 WiFi 网络。
  - 专用网络（例如公司或大学建立的内联网）要求只有提供正确的安全质询的用户才能访问网络。潜在用户在连接序列或通过安全的虚拟专用网络 (virtual private network, VPN) 应用程序登录到网络的过程中必须提供密钥。
- 加密与 WiFi 网络的连接。
  - 可以使用安全密钥对您的系统与 WiFi 网络之间的通信加密。您的 WiFi 网络访问点必须是您的家庭或办公室中具有安全密钥生成功能的路由器。在创建安全连接之前，您的系统和路由器首先建立密钥，然后共享密钥。

dladm 命令可以使用有线对等保密 (Wired Equivalent Privacy, WEP) 或 Wi-Fi 保护访问 (Wi-Fi Protected Access, WPA) 密钥对通过访问点的连接加密。WEP 协议在针对无线连接的 IEEE 802.11 规范中定义。WPA 协议在针对无线连接的 IEEE 802.11i 规范中定义。Oracle Solaris 支持 WPA 标准的版本 1 和 2。有关与 WEP 和 WPA 相关的 dladm 命令选项的更多信息，请参阅 [dladm\(1M\)](#) 手册页。

### ▼ 如何通过指定 WEP 密钥来设置加密 WiFi 网络连接

以下过程说明如何设置系统与家用路由器之间的安全通信。许多家用无线和有线路由器具有加密功能，能够生成安全密钥。

**开始之前** 如果您要连接到家庭无线网络，请确保已配置路由器且已生成 WEP 密钥。按照路由器制造商的文档生成和保存密钥配置。

1. 成为管理员。
2. 如下所示创建一个包含 WEP 密钥的安全对象：

```
# dladm create-secobj -c wep keyname
```

其中 *keyname* 代表您要给予密钥的名称。

3. 将 WEP 密钥的值提供给安全对象。  
然后，`create-secobj` 子命令运行一个请求密钥值的脚本。

```
provide value for keyname: 5-or-13-byte key
confirm value for keyname: Retype key
```

此值就是路由器生成的密钥。该脚本接受一个包含 5 个字节或 13 个字节的字符串，它是密钥值的 ASCII 格式或十六进制格式。

4. 查看您刚刚创建的密钥的内容。

```
# dladm show-secobj
OBJECT          CLASS
keyname         wep
```

其中 *keyname* 是安全对象的名称。

5. 建立与 WiFi 网络的加密连接。

```
# dladm connect-wifi -e network -k keyname interface
```

6. 验证连接是安全的。

```
# dladm show-wifi
LINK    STATUS    ESSID    SEC    STRENGTH    MODE    SPEED
net0    connected  wifi-1  wep    good        g       11Mb
```

在以上输出中，位于 `SEC` 列下的 `wep` 值指示连接具有 WEP 加密。

#### 例 29 使用 WEP 密钥设置加密 WiFi 通信

以下示例假定您已经完成下面的内容：

- 按照路由器制造商的文档创建了 WEP 密钥。
- 保存了密钥，以便使用它在您的系统上创建安全对象。

如下所示创建安全对象：

```
# dladm create-secobj -c wep mykey
provide value for mykey: *****
confirm value for mkey: *****
```

当您提供由路由器生成的 WEP 密钥时，您键入的值显示为星号。

以下命令通过使用安全对象 `mykey` 建立到 WiFi 网络 `citinet` 的加密连接。

```
# dladm show-secobj
OBJECT          CLASS
mykey           wep
# dladm connect-wifi -e citinet -k mykey net0
```

以下命令验证您是否已通过 WEP 加密连接到 citinet 无线网络。

```
# dladm show-wifi
LINK          STATUS      ESSID          SEC          STRENGTH      MODE      SPEED
net0         connected  citinet       wep          good          g         36Mb
```

## 在反应性模式下管理已知 WLAN

如果您正在使用反应性配置文件，则还可以利用系统自动连接到已知 WLAN 的功能。连接到 WLAN 时，有关该 WLAN 的信息保留在 known-wlan 类型的网络配置文件中。还可以使用 netcfg 命令手动创建 known-wlan 配置文件。有关已知 WLAN 的概述信息，请参见[“已知 WLAN 的说明” \[89\]](#)。有关与任务相关的信息，请参见[“创建已知 WLAN” \[106\]](#)。

## 从桌面管理无线网络

缺省情况下，启用了无线网络连接时，反应性网络守护进程 (nwmamd) 会尝试在不询问的情况连接到收藏列表中的任何可用网络，采用的优先级顺序与这些连接的列出顺序相同。如果没有收藏的网络可用，将打开 "Wireless Chooser" (无线选择器) 对话框。在此对话框中，可以选择要连接的无线网络。

在网络管理 GUI "Network Preferences" (网络首选项) 对话框的 "Connection Properties" (连接属性) 视图中，可在 "Wireless" (无线) 选项卡内修改尝试连接到无线网络的方法。如果需要，您还可以通过右键单击位于桌面上的网络状态通知图标，手动连接到不同的无线网络。

---

提示 - 可以通过 "Network Preferences" (网络首选项) 对话框访问选定网络的 "Connection Properties" (连接属性) 视图。此对话框包含一个名为 "Show" (显示) 的下拉式列表。此列表使您能够为给定网络切换视图。在每个视图中，存在您可以执行的各种任务和关于所选网络的特定于该视图的信息。

系统中的各网络配置文件中的每一个网络连接都有以下视图：

- 连接状态
- 网络配置文件
- 连接属性

有关基于配置文件的网络配置的更多信息，请参见[第 5 章 关于在 Oracle Solaris 中管理基于配置文件的网络配置](#)和 GUI 联机帮助。

---

## ▼ 如何连接无线网络

右键单击 "Network Status" (网络状态) 通知图标后将显示 "Join Wireless Network" (连接无线网络) 选项, 通过选择该选项连接无线网络。"Wireless Chooser" (无线选择器) 对话框中显示可供连接的无线网络列表。

1. 要手动连接到其他无线网络, 请执行以下操作之一:
  - 从 "Network Status" (网络状态) 通知图标的右键菜单中选择一个可用无线网络。
  - 从 "Network Status" (网络状态) 通知图标菜单中选择 "Join unlisted wireless network" (连接未列出的无线网络) 选项。  
未列出的无线网络是指, 配置为不广播其网络名称但仍然可以连接的无线网络。
  - 从 "Wireless Chooser" (无线选择器) 对话框中选择一个可用无线网络。当可以选择可用无线网络进行连接时, 将自动显示此对话框。



2. 如果 "Join Wireless Network" (连接无线网络) 对话框打开, 请为选择的无线网络提供所需的所有信息。

## 从桌面管理收藏的无线网络

缺省情况下, 首次连接某个无线网络时, 在 "Join Wireless Network" (连接无线网络) 对话框中将显示 "Add to list of favorite networks on successful connection" (成功连接时添加到收藏的网络列表中) 复选框。



如果要在连接成功时将无线网络添加到您的收藏列表中，请选中此框。如果不希望将网络添加到收藏列表中，请取消选中此框。缺省情况下选中此框。

要添加当前不可用或当前未将其网络名称广播到收藏列表中的无线网络，请转到 "Connection Properties" (连接属性) 视图中的 "Wireless" (无线) 选项卡，然后单击 "Add" (添加) 按钮。要添加网络，您需要知道其网络名称、安全类型和安全密钥。

# 索引

---

## A

- 安全和授权, 92
- 安全注意事项
  - WiFi, 128
- activation-mode 属性, 91
  - 不同配置文件类型的属性值, 92
- autopush 属性, 32

## B

- 半双工, 31
- 包转发
  - 在接口上, 63
- 本地地址, 41
- BSSID, 124

## C

- 创建和管理反应性配置文件, 85
- 创建配置文件
  - 以交互方式创建
    - 具有 NCU 的 NCP, 101
    - 已知 WLAN 配置文件, 106
  - 以交互方式创建 ENM, 105
- 创建
  - WLAN 创建, 106
  - 为 NCP 创建 NCU, 100
  - 位置配置文件, 103
  - 创建 ENM 的配置文件
    - ENM 配置文件, 105
- 创建网络配置文件
  - ENM 创建, 105
- cfgadm 命令, 36
- CIDR 表示法, 41

## D

- 导出和恢复配置文件配置, 117
- 地址
  - 临时, 在 IPv6 中, 46
- 地址对象, 42
- 地址解析协议 (Address Resolution Protocol, ARP), 40
- 电源管理, 31
- 顶级配置文件, 97
- 动态路由
  - 最佳用途, 53
- 动态重新配置 (dynamic reconfiguration, DR)
  - 更换 NIC, 35
- 对称路由, 62
- 多宿主主机, 62
  - 定义, 59
  - 针对 IPv6 启用, 45
- DefaultFixed 位置, 108
- DHCP, 42
- dladm 命令, 15, 26
  - delete-phys, 28
  - rename-link, 29
  - reset-linkprop, 29
  - scan-wifi, 124
  - set-linkprop, 29
  - show-ether, 32, 33
  - show-link, 27
  - show-linkprop, 32, 127
  - show-phys, 27
  - show-wifi, 125
- dlstat 命令, 29

## E

- ECMP, 62
- ENM

以交互方式创建 ENM 配置文件, 105

ESSID, 124

/etc/hosts 文件, 41

/etc/inet/ndpd.conf 文件

临时地址配置, 47

## F

反应性配置文件

配置文件激活策略, 91

反应性网络配置

删除配置文件, 116

导出配置文件配置, 119

反应性网络配置模式

网络配置文件和类型, 85

服务器, IPv6

启用 IPv6, 51

辅助状态值

显示, 116

## G

更改

使用 walkprop 子命令更改属性值, 113

配置文件的属性值, 108

## H

恢复配置文件, 119

获取特定属性的值, 112

## I

ICMP, 40

ifconfig 命令, 39

IP 地址

DHCP, 42

IPv4 和 IPv6, 41

删除, 66

包转发, 63

属性, 64, 71

本地和远程, 41

监视, 67

静态, 41

IP 接口

IP 地址, 70, 71

删除 IP 地址, 66

删除接口配置, 65

启用包转发, 63

地址属性, 64

指定 IP 地址, 41

接口属性, 69

显示

IP 地址, 70

地址属性, 71

常规信息, 42, 67

接口, 68

接口属性, 69

更改 IP 地址, 66

更改主接口, 65

监视, 67, 68

禁用和启用, 66

配置, 43

验证 MAC 地址的唯一性, 37

IP 隧道, 41

本地和远程地址, 41

ipadm 命令, 15

create-addr, 41

create-ip, 40

delete-addr, 66

delete-ip, 65

disable-if, 66

set-addrprop, 64

show-addr, 70

show-addrprop, 64, 71

show-if, 68

show-ifprop, 69

多宿主主机, 60

IPv6

临时地址配置, 46

启用, 在服务器上, 51

路由, 59

## J

交互模式 netcfg 命令, 97

接口

配置

临时地址, 46

手动, 针对 IPv6, 45

- 接口 ID
  - 使用手动配置的标记, 51
- 禁用配置文件, 96
- 静态路由
  - 在主机上手动配置, 56
  - 最佳用途, 53
  - 添加静态路由, 54
  - 配置示例, 55
- 巨型帧
  - 启用支持, 30
- L
- 链路本地地址
  - 手动配置, 使用标记, 51
- 链路聚合, 26
- 链路名称, 29
  - 通用, 22
- 链路速度, 31
- 列出
  - 所有配置文件, 110
  - 特定属性的值, 112
  - 特定配置文件的所有属性值, 110
  - 系统中的配置文件信息, 110
- 临时地址, 在 IPv6 中
  - 定义, 46
  - 配置, 47
- 路由
  - IPv6, 59
  - 在单接口主机上, 56
  - 配置静态的, 56
- 路由表
  - 手动配置, 54
- M
- MAC 地址
  - 验证唯一性, 37
- MTU, 30
- N
- name-service/switch 服务, 41
- NCP
  - 创建具有 NCU 的 NCP
  - NCU 创建, 101
- NCU
  - 创建, 100
- ndpd.conf 文件
  - 临时地址配置, 47
- netadm 命令, 15
- netcfg 命令, 15
  - walkprop 子命令, 113
  - 交互模式, 97
- netstat 命令
  - 检查 WiFi 链路上的包流, 127
- nis/tdomain SMF 服务
  - 本地文件模式配置, 77
- P
- 配置
  - 手动配置接口, 针对 IPv6, 45
- 配置文件
  - 网络配置文件类型, 85
  - 配置文件激活策略, 91
- Q
- 启用配置文件, 96
- 全双工, 31
- R
- route 命令, 15, 43
- S
- 删除配置文件, 116
- 设置和更改配置文件的属性值, 108
  - 以交互方式设置属性值, 109
  - 在命令行模式下设置属性值, 109
- 示例
  - 以交互方式设置属性值, 109
  - 启用配置文件, 96
  - 导出配置文件配置, 118
  - 在命令文件模式下, 119
  - 禁用配置文件, 96
- 属性
  - activation-mode 属性, 91
  - 以交互方式获取和列出单个属性值, 112

- 以交互方式设置属性值, 108
- 使用 walkprop 子命令更改属性值, 113
- 列出特定配置文件的所有属性值, 110
- 获取特定属性的值, 112
- 设置和更改配置文件的属性值, 108
- 数据链路
  - autopush 属性, 32
  - STREAMS 模块, 32
  - VLAN, 26
  - VNIC, 26
  - 以太网参数值, 32
  - 删除, 28
  - 显示
    - 在系统上的物理位置, 28
    - 常规信息, 26
    - 物理属性, 27
    - 网络驱动程序属性, 32, 33
    - 链路, 27
    - 链路属性, 32
  - 更改 MTU 大小, 30
  - 物理链路, 26
  - 自动协商, 30
  - 获取运行时统计信息, 29
  - 设置属性, 29
  - 通告速度和支持的速度, 31
  - 通用名称, 29
  - 重命名, 29
  - 针对使用定制名称的规则, 25
  - 链路名称, 24
  - 链路聚合, 26
  - 链路速度, 31
- SCTP, 40
- STREAMS 模块
  - 和数据链路, 32
- U
  - UDP, 40
- W
  - 网络接口卡 (network interface card, NIC)
    - 通过 DR 进行更换, 35
  - 网络配置
    - 启用了 IPv6 的多宿主主机, 45
  - 网络配置工具
    - dladm 命令, 26
    - 网络配置命令, 15
    - 网络配置授权
      - 和安全性, 92
    - 网络配置文件
      - ENM, 86
      - NCU, 86
      - 位置配置文件, 86
      - 列出系统中的配置文件信息, 110
      - 创建
        - 位置配置文件, 103
      - 创建和管理配置文件, 85
      - 删除配置文件, 116
      - 导出和恢复配置文件配置, 117
      - 已知 WLAN, 87
      - 恢复导出的配置文件, 119
      - 设置和更改配置文件的属性值, 108
    - 网络配置文件配置
      - 列出系统中的配置文件信息, 110
      - 创建
        - 位置配置文件, 103
      - 导出和恢复配置文件配置, 117
      - 设置和更改配置文件的属性值, 108
    - 网络拓扑
      - 自治系统, 57
    - 位置配置文件
      - 以交互方式创建, 103
      - 创建位置配置文件, 103
    - 无线接口, 123
    - walkprop 子命令
      - 查看和更改属性值, 113
    - WiFi
      - IEEE 802.11 规范, 123
      - WiFi 配置示例, 126
      - 加密的通信示例, 129
      - 加密连接, 128
      - 基本服务集 ID (Basic Service Set ID, BSSID), 124
      - 安全的 WiFi 链路, 128
      - 定义, 123
      - 扩展服务集 ID (Extended Service Set ID, ESSID), 124
      - 检查包流, 127
      - 监视链路, 127
      - 示例, 设置链路速度, 128
      - 连接到 WiFi 网络, 123

## X

## 显示

- 辅助状态值, 116

## 新增功能

- IPv6 中的临时地址, 46

- 手动配置链路本地地址, 49

- 虚拟局域网 (virtual local area network, VLAN), 26

- 虚拟网卡 (virtual network card, VNIC), 26

- 虚拟专用网络 (virtual private network, VPN), 32

## Y

- 已知 WLAN, 106

- 已知 WLAN 配置文件

- 以交互方式创建, 106

- 以交互方式创建

- ENM 配置文件, 105

- 具有 NCU 的 NCP, 101

- 已知 WLAN 配置文件, 106

- 以交互方式获取单个属性值, 112

- 以交互方式设置属性值, 108

- 以太网参数, 32

## 域名

- nis/domain SMF 服务, 77

- 远程地址, 41

- 运行时统计信息

- 数据链路

- dlstat, 29

## Z

## 主机

- 临时 IPv6 地址, 46

- 多宿主

- 配置, 59

- 主接口, 切换, 29, 34, 65

- 自动协商, 30

- 自治系统 (autonomous system, AS) 见 网络拓扑

